

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-231070

(43)Date of publication of application : 05.09.1997

(51)Int.Cl.

G06F 9/30
G06F 9/30
G06F 9/305

(21)Application number : 08-353167

(71)Applicant : SEIKO EPSON CORP

(22)Date of filing : 16.12.1996

(72)Inventor : KUBOTA SATORU
KUDO MAKOTO
MIYAYAMA YOSHIYUKI

(30)Priority

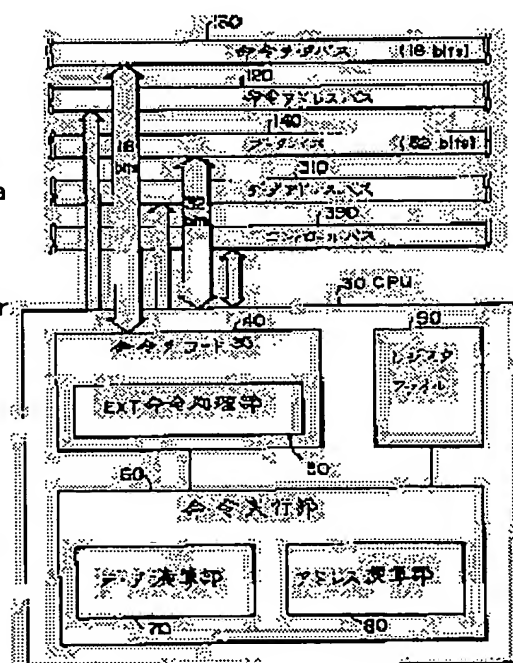
Priority number : 07332207 Priority date : 20.12.1995 Priority country : JP

(54) INFORMATION PROCESSING CIRCUIT, SEMICONDUCTOR INTEGRATED CIRCUIT DEVICE, MICROCOMPUTER AND ELECTRONIC EQUIPMENT

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a function for extending an immediate value included in an instruction code as needed without complicating a control circuit by extending the immediate value required for the execution of a target instruction based on a prefix instruction.

SOLUTION: An instruction decoding part 40 is for decoding an inputted instruction code and performing a processing required for executing the instruction and is provided with an ext instruction processing part 50. An instruction execution part 60 is for executing the instruction based on the operation contents of the instruction analyzed by the instruction decoding part 40 and is provided with a data operation part 70 and an address operation part 80. The instruction code processed by a CPU 30 is the fixed length of 16 bits and the instruction code of the fixed length 16 bits is stored in an instruction code storage part connected through an instruction data bus and an instruction address bus.



Then, by newly providing the instruction of the fixed length 16 bits which is an ext instruction and using the prefix instruction, the bit width of the immediate value is extended.

LEGAL STATUS

[Date of request for examination] 14.11.2002

[Date of sending the examiner's decision of rejection] 01.06.2004

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection] 2004-13618

[Date of requesting appeal against examiner's decision of rejection] 01.07.2004

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

*** NOTICES ***

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] It is the information processing circuit where a given target instruction and the prefix instruction for extending this target instruction function are inputted. An instruction code analysis means to input said target instruction and a prefix instruction, and to analyze the contents of operation of this instruction code, An instruction-execution means to execute an instruction based on the contents of operation which said instruction code analysis means analyzed is included. Said instruction code analysis means It is constituted so that an immediate escape means to extend an immediate required for activation of the target instruction set as the object of this prefix instruction-function escape may be included based on said prefix instruction. Said instruction-execution means is an information processing circuit characterized by executing a target instruction based on the immediate extended by said immediate escape means.

[Claim 2] The information processing circuit characterized by being constituted so that fixed-length instruction code may be inputted and performed in claim 1.

[Claim 3] The information processing circuit characterized by the bit width of face of said fixed-length instruction code being the data which can process said information processing circuit, or below the bit width of face of the address in claim 2.

[Claim 4] It is the information-processing circuit characterized by to set to claim 3, to be constituted said information-processing circuit so that the fixed-length instruction code of the bit width of face of 16 may be inputted, and for said immediate escape means to extend an immediate required for activation of said target instruction to width of face of 32 bits based on said prefix instruction, and for said instruction-execution means to include an arithmetic logical-operation means execute a target instruction using the immediate of 32 bits extended with said immediate escape means.

[Claim 5] It is the information processing circuit characterized by extending the immediate by which said immediate escape means is included in the instruction code of said target instruction in either claim 1 - claim 4 based on the immediate of the instruction code of a prefix instruction, and the immediate of the instruction code of said target instruction.

[Claim 6] It is the information processing circuit characterized by extending the data used for activation of a target instruction using the immediate by which said immediate escape means was included in the instruction code of a prefix instruction in either claim 1 - claim 5.

[Claim 7] It is the information processing circuit characterized by extending the address used for activation of a target instruction using the immediate by which said immediate escape means was included in the instruction code of a prefix instruction in either claim 1 - claim 6.

[Claim 8] It is the information processing circuit characterized by extending an immediate required for activation of the target instruction set as the object of this prefix instruction-function escape using each immediate by which said immediate escape means was included in the instruction code of two or more prefix instructions in either claim 1 - claim 7.

[Claim 9] It is the information processing circuit characterized by including the common code for distinguishing that each instruction code of said prefix instructions of two or more is a prefix instruction in claim 8, respectively.

[Claim 10] In either claim 8 or claim 9 said immediate escape means A processing-state storage means to memorize the processing state determined based on the count of a continuation input of a prefix instruction, Said data-hold means to hold the immediate of the inputted prefix instruction based on the processing state memorized by said processing-state storage means to the 1st register – the m-th register (for m to be one or more integers), The information processing circuit characterized by including a means to extend and generate an immediate required for activation of an instruction of said target instruction, based on the immediate held at the 1st register of said data-hold means – the m-th register (m is one or more integers).

[Claim 11] Said instruction-execution means is the information-processing circuit characterized by to execute this target instruction from the contents of operation extended with said target instruction expansion means including a target instruction expansion means extend and interpret the contents of operation of this target instruction in either claim 1 – claim 10 when said instruction code analysis means inputs a target instruction after a prefix instruction input.

[Claim 12] It is the information processing circuit characterized by what said target instruction expansion means extends the contents of operation of this target instruction to 3 operand instructions in claim 11 using the immediate contained in two operands and prefix instructions of this target instruction when the predetermined target instruction which is 2 operand instructions was inputted after a prefix instruction input, and is interpreted.

[Claim 13] it be the information processing circuit characterize by to extend the contents of operation so that said target instruction expansion means correct the data or the address stored in the register specified as the instruction code of a target instruction based on the immediate contained in the instruction code of a prefix instruction when the predetermined target instruction which have a register assignment value after a prefix instruction input be inputted in claim 11 and it may perform .

[Claim 14] It is the information-processing circuit characterize by to extend the contents of operation so that the displacement of the address stored in the register specified as the instruction code of a target instruction based on the immediate contained in the instruction code of a prefix instruction creates and it may perform using this displacement when the predetermined target instruction with which said target instruction expansion means has a register assignment value after a prefix instruction input in claim 13 is input .

[Claim 15] It is the information processing circuit characterized by including a prefix instruction expansion means by which said instruction code analysis means extends the contents of operation of a prefix instruction in either claim 1 – claim 14 based on the count of a continuation input of a prefix instruction.

[Claim 16] The information processing circuit characterized by containing in the instruction code storage means pan which memorizes the instruction code inputted into said instruction code analysis means in either claim 1 – claim 15.

[Claim 17] Semiconductor integrated circuit equipment characterized by including an information processing circuit according to claim 1 to 16.

[Claim 18] The microcomputer characterized by including an information processing circuit according to claim 1 to 17.

[Claim 19] The microcomputer characterized by being a RISC method in claim 18.

[Claim 20] Electronic equipment characterized by controlling using a microcomputer according to claim 18 or 19.

[Translation done.]

*** NOTICES ***

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the electronic equipment constituted using an information processing circuit and semiconductor integrated circuit equipment, the microcomputer that contains said information processing circuit, and this microcomputer.

[0002]

[Background of the Invention] Conventionally, in the microcomputer of a RISC method which can process 32-bit data, the instruction code fixed to 32-bit width of face is used. The reason is that the time amount which decoding of an instruction takes can be shortened compared with the case where the instruction code of variable-length bit width of face is used, and it can make the circuit scale of a microcomputer small if the instruction code of fixed-length bit width of face is used.

[0003] However, also in a 32-bit microcomputer, there is also much instruction code which is not needed especially 32 bits. Therefore, if 32 bits describes the instruction code of all instructions, the instructions which a redundant part produces in instruction code will increase in number, and the utilization ratio of memory will worsen.

[0004] In such a case, it is also possible to execute an instruction, carrying out the logic reduction of the redundant instruction code, and decoding to the original instruction inside a microcomputer. However, there was a problem that a control circuit became complicated, by such method.

[0005]

[Problem(s) to be Solved by the Invention] then, this invention — the person was performing examination about the microcomputer which processes the fixed-length instruction code of bit width of face shorter than the bit width of face of the data which can be processed in order to raise the utilization ratio of memory, without complicating a control circuit.

[0006] However, if 32 bit fixed length's instruction code is only made into a 16-bit fixed length, for example, the following troubles will arise.

[0007] Usually, in order to also describe operands other than an operation code or an immediate to instruction code, even if it uses 16-bit instruction code, the number of bits of the immediate which can be used will become still smaller than 16 bits. That is, in spite of being able to process 32-bit data, the problem that only an immediate smaller than 16 bits can be specified by instruction code arises.

[0008] Moreover, it is difficult to, secure the field of the operand of 3 operand instructions by 16-bit instruction code for example. Therefore, the problem how to process the operation that description is difficult, by short instruction code to this appearance arises.

[0009] In order to solve the above-mentioned problem, the function which extends the immediate contained in short instruction code if needed is needed. Moreover, in order to perform operation which cannot be described by short instruction code, the function which extends the contents of operation is needed.

[0010] The purpose of this invention is offering the information processing circuit which has the function which extends the immediate contained in instruction code if needed, semiconductor integrated circuit equipment, a microcomputer, and electronic equipment, without complicating a control circuit.

[0011] Moreover, other purposes of this invention are offering the information processing circuit which has the function which extends the contents of operation, semiconductor integrated circuit equipment, a microcomputer, and electronic equipment, without complicating a control circuit.

[0012]

[Means for Solving the Problem] In order to attain said purpose, invention of claim 1 It is the information processing circuit where a given target instruction and the prefix instruction for extending this target instruction function are inputted. An instruction code analysis means to input said target instruction and a prefix instruction, and to analyze the contents of operation of this instruction code, An instruction-execution means to execute an instruction based on the contents of operation which said instruction code analysis means analyzed is included. Said instruction code analysis means It is constituted so that an immediate escape means to extend an immediate required for activation of the target instruction set as the object of this prefix instruction-function escape may be included based on said prefix instruction. Said instruction-execution means is characterized by executing a target instruction based on the immediate extended by said immediate escape means.

[0013] An instruction code analysis means decodes the inputted instruction code, and performs processings required in order that an instruction-execution means may execute an instruction, such as calculating the address of the storage means which is set as the object of an instruction and by which the data storage is carried out.

[0014] The prefix instruction which extends a target instruction and this target instruction function is inputted into the information processing circuit of this invention. A prefix instruction has the function which extends the target instruction function, in case [its] it does not perform in an instruction-execution means but a target instruction of consecutiveness is executed, if independent.

[0015] An immediate escape means extends an immediate required for activation of a target instruction based on a prefix instruction. Immediates required for activation of a target instruction are the immediate contained in the instruction code of a target instruction for example, and an immediate contained in the instruction code of a prefix instruction, and the immediate which is needed in case it performs is said.

[0016] With the escape of an immediate here, the bit width of face of the immediate indicated in instruction code besides a usual zero escape and a usual sign escape is elongated, and also when compensating a given bit, it contains in the elongated part. That is, according to this invention, based on a prefix instruction, said given bit can be compensated and an immediate can be extended.

[0017] Therefore, it becomes possible to set up short the number of bits of the instruction code containing the immediate used as the cause which enlarges the number of bits of instruction code. For this reason, even if it does not have a complicated control circuit for decoding variable-length instruction code, the utilization ratio of memory is improvable with the easy configuration which is short fixed-length instruction code of adopting.

[0018] Invention of claim 2 is characterized by being constituted so that fixed-length instruction code may be inputted and performed in claim 1.

[0019] According to this invention, by using fixed-length instruction code, the time amount which decoding of an instruction takes can be shortened compared with the case where the instruction code of variable-length bit width of face is used, and the circuit scale of an information processing circuit can be made small.

[0020] Moreover, with a prefix instruction, since an immediate is extensible, the immediate used as the cause which lengthens instruction code can be set up short. Therefore, since the die length of fixed-length instruction code can be set up short, parts with redundant instruction code can be reduced efficiently, and the utilization ratio of memory, especially the memory which memorizes instruction code can be improved.

[0021] Invention of claim 3 is characterized by the bit width of face of said fixed-length instruction code being the data which can process said information processing circuit, or below the bit width of face of

the address in claim 2.

[0022] According to this invention, even if it makes bit width of face of instruction code into the small bit width of face below the data which can process an information processing circuit, or the bit width of face of the address, it is extensible to the bit width of face which can process said information processing circuit in an immediate. Thus, since bit width of face of instruction code is made to the data which can process an information processing circuit, or below the bit width of face of the address, the utilization ratio of memory is improvable.

[0023] Invention of claim 4 is set to claim 3. Said information processing circuit It is constituted so that the fixed-length instruction code of the bit width of face of 16 may be inputted. Said immediate escape means Based on said prefix instruction, an immediate required for activation of said target instruction is extended to width of face of 32 bits, and it is characterized by said instruction-execution means including an arithmetic logical operation means to execute a target instruction using the immediate of 32 bits extended with said immediate escape means.

[0024] Conventionally, in the microcomputer of a RISC method which can process 32-bit data, processing was performed using the instruction code of 32 bits of fixed lengths. However, especially in said instruction code, there was also much instruction code which is not needed 32 bits, and it caused this utilization ratio aggravation of memory.

[0025] That is, instruction code usually consists of the class code and operation code which define the fundamental operation of this instruction, and operands other than an immediate. In here, operands other than an immediate put the code corresponding to a source register (rs) and a destination register (rd) etc. For this reason, when using 16-bit instruction code, the number of bits of the immediate which can be used will become still smaller, and will usually become about 6 bits. Generally, reservation of the bit width of face of an immediate becomes difficult, so that instruction code is short. However, according to this invention, since an immediate is extensible, also in the information processing circuit which can process the data which are 32 bits, the short instruction code of 16 bits can be used. For this reason, utilization ratio large ** of memory is improvable.

[0026] Invention of claim 5 is characterized by said immediate escape means extending the immediate contained in the instruction code of said target instruction based on the immediate of the instruction code of a prefix instruction, and the immediate of the instruction code of said target instruction in either claim 1 – claim 4.

[0027] When it is made this appearance, an immediate can be extended with the easy configuration of specifying a part of immediate required for activation of a target instruction as an immediate of the instruction code of a target instruction, and specifying the remaining part of an immediate required for activation of a target instruction as an immediate of the instruction code of a prefix instruction to extend and perform the immediate contained among the instruction code of a target instruction.

[0028] Therefore, it becomes possible to carry out a short **** setup of the number of bits for the immediates in instruction code; and the utilization ratio of memory can be improved.

[0029] Invention of claim 6 is characterized by said immediate escape means extending the data used for activation of a target instruction using the immediate contained in the instruction code of a prefix instruction in either claim 1 – claim 5.

[0030] In here, data are a concept contrasted with the address and the contents of the value put what expresses except the address. The data used for activation of a target instruction mean the data of the immediate contained in the instruction code of a target instruction, and the data of the immediate used in case it performs although not contained in the instruction code of a target instruction. If it is made this appearance, a target instruction can be executed using the immediate of the data extended with the easy configuration.

[0031] Invention of claim 7 is characterized by said immediate escape means extending the address used for activation of a target instruction using the immediate contained in the instruction code of a prefix instruction in either claim 1 – claim 6.

[0032] In here, the address is a concept contrasted with data and information for the contents of the value to show a certain specific location and specific location in storage is said. The address of an immediate with which the address used for activation of a target instruction is included in the instruction code of a target instruction, and the address of an immediate used in case it performs although not contained in the instruction code of a target instruction are said. If it is made this appearance, a target instruction can be executed using the immediate of the address extended with the easy configuration.

[0033] Invention of claim 8 is characterized by said immediate escape means extending an immediate required for activation of the target instruction set as the object of this prefix instruction-function escape using each immediate contained in the instruction code of two or more prefix instructions in either claim 1 – claim 7.

[0034] thus — if it carries out — the immediate of two or more prefix instructions — group bubble **** — an immediate required for activation of a target instruction is extensible to arbitration with things. Therefore, since the activation using a big immediate can be treated even if it uses short instruction code, the evil by compaction of instruction code can be prevented.

[0035] Invention of claim 9 is characterized by including the common code for distinguishing that each instruction code of said prefix instructions of two or more is a prefix instruction, respectively in claim 8.

[0036] According to this invention, there is no need of preparing two or more different prefix instructions. For this reason, also in the information processing circuit which processes instruction code with little number of bits, it becomes extensible [an immediate], without omitting the instruction of other classes.

[0037] Invention of claim 10 is set to either claim 8 or claim 9. Said immediate escape means A processing-state storage means to memorize the processing state determined based on the count of a continuation input of a prefix instruction, Said data-hold means to hold the immediate of the inputted prefix instruction based on the processing state memorized by said processing-state storage means to the 1st register – the m-th register (for m to be one or more integers), It is characterized by including a means to extend and generate an immediate required for activation of an instruction of said target instruction, based on the immediate held at the 1st register of said data-hold means – the m-th register (m is one or more integers).

[0038] The processing state determined by the count of a continuation input itself is sufficient as the processing state determined based on the count of a continuation input of a prefix instruction in here, and the processing state uniquely drawn from the count of a continuation input according to a certain specific Ruhr is sufficient as it. For example, when a prefix instruction is ***** (ed) 3 times or more and inputted, 3rd henceforth may be constituted so that it may be in the always same processing state. And with reference to the processing state determined by carrying out in this way, the immediate of a prefix instruction can be held for an immediate maintenance means, and an immediate can be extended: by combining the this held immediate. If it is made this appearance, based on the count of an input which the prefix instruction followed, an immediate required for activation of a target instruction is extensible with an easy configuration.

[0039] Said instruction-execution means is characterized by to execute this target instruction from the contents of operation extended with said target instruction expansion means including a target instruction expansion means to by which invention of claim 11 extends and interprets the contents of operation of this target instruction when said instruction code analysis means inputs a target instruction after a prefix instruction input in either claim 1 – claim 10.

[0040] If it is made this appearance, the function extended to the target instruction can be realized by performing a target instruction combining a prefix instruction. For this reason, the number of instructions can be reduced and it becomes possible to reduce the number of bits further used for instruction code.

[0041] Moreover, in instruction code with little number of bits, it also becomes possible to realize the

function that implementation is difficult, by combining a prefix instruction and a target instruction. For this reason, the utilization ratio of the memory by compaction of instruction code is improvable.

[0042] Invention of claim 12 is characterized by extending the contents of operation of this target instruction to 3 operand instructions, and interpreting them using the immediate by which said target instruction expansion means was included in two operands and prefix instructions of this target instruction when the predetermined target instruction which is 2 operand instructions was inputted after a prefix instruction input, in claim 11.

[0043] If it carries out such, by performing a target instruction of 2 operand instructions combining a prefix instruction, the actuation function of this target instruction can be extended to 3 operand instructions, and can be performed. Therefore, even if it uses the short instruction code which has only a description field by two operands, 3 operand instructions are realizable. For this reason, it becomes possible to reduce the number of bits used for the number of instructions, and instruction code.

[0044] Furthermore, since immediate data can be extended with a prefix instruction and 3 operand instructions can be realized, compared with the case where 3 operand instructions are executed, big immediate data can usually be treated with one instruction.

[0045] Invention of claim 13 is characterized by to extend the contents of operation so that said target instruction expansion means corrects the data or the address stored in the register specified as the instruction code of a target instruction based on the immediate contained in the instruction code of a prefix instruction when the predetermined target instruction which has a register assignment value after a prefix instruction input was inputted in claim 11 and it may perform .

[0046] In here, it says performing four operations or logical operation, shift operation, etc. to data or the address as correcting and performing the address of data, and performing. If it carries out such, since the data or the address stored in the register specified by the instruction code of a target instruction by performing a target instruction combining a prefix instruction can be corrected and performed by the immediate contained in the instruction code of a prefix instruction, it becomes possible to reduce the number of bits used for the number of instructions, or instruction code.

[0047] When the predetermined target instruction with which said target instruction expansion means has a register assignment value after a prefix instruction input in claim 13 is inputted, invention of claim 14 creates the displacement of the address stored in the register specified as the instruction code of a target instruction, and is characterized based on the immediate contained in the instruction code of a prefix instruction by to extend the contents of operation so that it may perform using this displacement.

[0048] If it does in this way, it will become possible by using a prefix instruction to create the displacement of the address at the time of activation.

[0049] Furthermore, addition of displacement is attained by using a prefix instruction, without being limited to the number of bits of an operand.

[0050] Invention of claim 15 is characterized by said instruction code analysis means including a prefix instruction expansion means to extend the contents of operation of a prefix instruction, based on the count of a continuation input of a prefix instruction in either claim 1 – claim 14.

[0051] In here, when there is a prefix instruction of two or more classes, the contents of operation of a prefix instruction are extended based on the count of a continuation input of a prefix instruction of the same kind. A prefix instruction of the same kind [here] means the case where the contents of operation of this prefix instruction are the same. therefore, the case where there is a prefix instruction of two or more classes — it is — those functions — an operation code etc. — things — when like, what has the same operation code etc. is called prefix instruction of the same kind.

[0052] If it carries out such, since the contents of operation of a prefix instruction can be made to change based on the count of an input, many functions are realizable with the prefix instruction of few classes.

[0053] Invention of claim 16 is characterized by containing in the instruction code storage means pan which memorizes the instruction code inputted into said instruction code analysis means in either claim

1 - claim 15.

[0054] If it is made this appearance, an information processor with the sufficient utilization ratio of memory in which instruction code with few [it is short and] redundancy parts was stored is realizable. Therefore, if it is the same magnitude, the information processor which can store more instructions can be offered, and if it is the same function, it is realizable by small memory space.

[0055] Invention of claim 17 is semiconductor integrated circuit equipment characterized by including an information processing circuit according to claim 1 to 16.

[0056] According to this invention, highly efficient semiconductor integrated circuit equipment can be offered by small circuitry.

[0057] Invention of claim 18 is a microcomputer characterized by including an information processing circuit according to claim 1 to 17.

[0058] According to this invention, a microcomputer with the sufficient utilization ratio of memory can be offered with an easy configuration.

[0059] Invention of claim 19 is a microcomputer characterized by being a RISC method in claim 18.

[0060] By processing fixed-length short instruction code, the microcomputer of a RISC method shortens the time amount which decoding of an instruction takes, and aims at making the circuit scale of a microcomputer small. Therefore, according to this invention, the microcomputer of the RISC method which can realize these purposes easily can be offered.

[0061] Invention of claim 20 is electronic equipment characterized by controlling using a microcomputer according to claim 18 or 19.

[0062] According to this invention, since the good information processing circuit of the utilization ratio of memory is built in with the easy configuration, cheap and highly efficient electronic equipment can be offered.

[0063]

[Embodiment of the Invention] Hereafter, the gestalt of operation of this invention is explained based on a drawing.

[0064] (Gestalt 1 of operation)

1. Functional description drawing 1 of CPU is the functional block diagram of CPU of the gestalt of operation of this invention. Although this CPU30 treats the data of 32-bit width of face, it is constituted so that the instruction code which is 16 bits may be processed.

[0065] A book CPU 30 contains the instruction decoding section 40, the instruction-execution section 60, and a register file 90. And this CPU performs an exchange of the exterior and a signal through the 16-bit instruction data bus 130, the instruction address bus 120 for an instruction data access, the 32-bit data bus 140, the data address bus 310 for a data access, and the control bus 390 for a control signal.

[0066] Said instruction decoding section 40 is also performing processing required in order to decode the inputted instruction code and to execute this instruction, and contains the ext instruction-processing section 50 which is the characteristic function of this invention. The register file 90 has the register used by CPUs, such as 16 general-purpose registers of general-purpose registers R0-R15, a program counter (PC), a processor status register (PSR), a stack pointer (SP), an arithmetic low register (ALR), and an arithmetic high register (AHR). The instruction-execution section 60 executes this instruction based on the contents of operation of the instruction which said instruction decoding section 40 analyzed, and contains the data operation part 70 which calculates data, and the address-arithmetic section 80 which performs the operation of the address.

[0067] Since the data size of the hardware inside this CPU is 32 bits, all of the arithmetic logical operation performed in the data operation part 70 of said instruction-execution section 60 or the address-arithmetic section 80, the number of bits, a data transfer of a register of said register file, etc. become 32 bits. However, the instruction code which said CPU30 processes is a 16-bit fixed length. That is, the instruction code of 16 bits of fixed lengths is stored in the instruction code storage section

which is connected through the instruction data bus and the instruction address bus and which is not illustrated. Therefore, the number of bits of the effective immediate under 1 instruction turns into the number of bits except the number of bits which a class code and an operation code have among instruction codes. For this reason, the number of bits of the effective immediate under 1 instruction is restricted to the comparatively short bit width of face of 8 bits from 6 bits.

[0068] So, the instruction of 16 bits of fixed lengths called an ext instruction is newly prepared, and by using this instruction, it constitutes from a gestalt of this operation so that the bit width of face of an immediate can be made to extend. Moreover, by using an ext instruction, it constitutes from instruction code of short bit width of face called 16 bits of fixed lengths so that operation which cannot be described may be made possible.

[0069] Although an ext instruction does not perform its activation in CPU if independent at all, it is the prefix instruction which is an instruction which extends the target instruction function by Lycium chinense just before a target instruction. In addition, the instruction set as the object of the expansion by this prefix instruction is called target instruction.

[0070] The ext instruction-processing section 50 processes such an ext instruction, and it mentions later for details.

[0071] 2. Take for an example the semiconductor integrated circuit built in the configuration microcomputer of a semiconductor integrated circuit, and explain an example and activity of circuitry for realizing the characteristic function of this invention.

[0072] Drawing 2 illustrates a part required for explanation of the configuration of the semiconductor integrated circuit built in the microcomputer. This semiconductor integrated circuit 100 contains CPU30, ROM110, and RAM that is not illustrated. Said CPU30 performs an exchange of the exterior and a signal through said ROM110, RAM which is not illustrated, and the data bus 140 of 120 or 32 bits of instruction address buses for the 16-bit instruction data bus 130 and this instruction data access.

[0073] The 16-bit instruction code which described the processing which CPU30 is made to perform is memorized by said ROM110, and it functions on it as an instruction code storage means. And the instruction code of said prefix instruction or the instruction used as a target as well as the instruction code of the usual instruction is memorized by this ROM110.

[0074] Said CPU30 contains an instruction register 150, the instruction decoding circuit 160, the immediate generation circuit 170, EXT1 register 172, EXT2 register 174 and a register file 90, and ALU190.

[0075] An instruction register 150 stores the instruction code inputted through the instruction data bus 130 from said ROM110. The instruction decoding circuit 160 decodes the instruction code stored in the instruction register 150, and outputs a control signal required for activation. Moreover, the immediate in instruction code is started, and if needed, it holds to EXT1 register 172 and EXT2 register 174, or outputs to the immediate generation circuit 170.

[0076] The immediate generation circuit 170 generates the immediate extended to 32 bits using the immediate held at the immediate outputted from said instruction decoding circuit 160 and EXT1 register 172, and EXT2 register 174. That is, the instruction decoding circuit 160, the immediate generation circuit 170, EXT1 register 172, and EXT2 register 174 function as the instruction decoding section 40 shown in drawing 1, and the ext instruction-processing section 50.

[0077] ALU190 performs arithmetic operation, logical operation, and shift operation to the data of the value stored in the register of said extended immediate and register file 90, or the primary storage which is not illustrated. That is, ALU190 functions as data operation part 70 of the instruction-execution section 40 shown in drawing 1.

[0078] Next, the configuration which is the characteristic function of this invention and which generates the extended immediate is explained to a detail. Although the escape of the immediate of this invention has various patterns so that it may mention later, with the gestalt of this operation, the case where the escape which increases the number of bits of the immediate of a target instruction of consecutiveness

is performed using the immediate of a prefix instruction is taken for an example, and the detailed configuration is explained.

[0079] The instruction processed with this microcomputer can be divided into a usual instruction and a usual prefix instruction. About a predetermined instruction, it can become the target of a prefix instruction among the usual instructions (a target ***** instruction of a prefix instruction is called target instruction).

[0080] In this microcomputer, since 16-bit fixed-length instruction code is adopted, only 6 bits is securable as the field which specifies an immediate that it mentions later by the usual instruction. However, since the data length at the time of ALU190 grade performing is 32 bits, the immediate which is said immediate generation circuit 170 and was extended to 32 bits is generated, and it is inputted into ALU190. In such a case, there are the following two kinds as an escape of the usual bit. That is, they are the zero escape which makes zero altogether 26 bits of said 6-bit high orders, and the sign escape which extends the value of said 6-bit most significant bit to 26 bits of high orders. However, in such an escape, although CPU has 32-bit arithmetic proficiency, it cannot treat the immediate of the big number of bits which has meaningful contents. Therefore, with the gestalt of this operation, the configuration which can perform the escape which increases the number of bits of the immediate of a target instruction of consecutiveness is adopted using the immediate of a prefix instruction. With the gestalt of this operation, a means to hold the immediate contained in the instruction code of a prefix instruction in such an escape, and the means which increases the number of bits of an immediate based on the held immediate have realized.

[0081] A means to hold the immediate first contained in the instruction code of a prefix instruction is explained. As this means, the instruction decoding circuit 160 has the state machine which is not illustrated. The condition of this state machine changes by whether the instructions which the instruction decoding circuit 160 processed are whether it is a prefix instruction and the usual instruction. The instruction decoding circuit 160 makes the immediate of the processed prefix instruction hold to EXT1 register 172 or EXT2 register 174 based on the condition of a state machine. The immediate generation circuit 170 generates the extended immediate using the immediate held at said EXT1 register 172 or EXT2 register 174 based on the condition of a state machine. In addition, a latch circuit besides a register is sufficient as a maintenance means.

[0082] In addition, as mentioned later, the prefix instruction used with the microcomputer of the gestalt of this operation is one kind of ext instruction, and the instruction code of an ext instruction contains the immediate of 13 bits.

[0083] Drawing 3 is drawing showing transition of the condition of the state machine which the instruction decoding circuit 160 has. Said state machine can hold three kinds of conditions, S0, S1, and S2. S0 expresses the condition of having inputted the usual instruction which is not an ext instruction (prefix instruction), S1 expresses the condition of having inputted the 1st ext instruction (prefix instruction), and S2 expresses the condition of having inputted the 2nd ext instruction (prefix instruction).

[0084] As shown in this drawing, when the condition of a state machine is S0, when the usual instruction (ext=0) is inputted, a condition is still S0 (400), and if an ext instruction (ext=1) is inputted, a condition will change to S1 (410). Moreover, when the condition of a state machine is S1, if the usual instruction (ext=0) is inputted, a condition will change to S0 (420), and if an ext instruction (ext=1) is inputted, a condition will change to S2 (430). Furthermore, when the condition of a state machine is S2, and a condition will change to S0 if the usual instruction (ext=0) is inputted (440), and an ext instruction (ext=1) is inputted, a condition is still S2 (450).

[0085] That is, when the usual instruction was inputted, it changes into the condition of S0 and an ext instruction is inputted twice or more continuously, said state machine is constituted so that the condition of S2 may be held, until it inputs the usual instruction again.

[0086] Drawing 4 is the flow chart Fig. having shown the algorithm of the actuation which the instruction

decoding circuit 160 performs about maintenance of an immediate.

[0087] First, the instruction decoding circuit 160 decodes the instruction code inputted into the instruction register, and performs a condition setup of a state machine based on the current condition of the inside of the paddle this whose instruction code is an ext instruction, and a state machine (step 10).

[0088] When the condition of a state machine is S1, it is in (step 20) and the condition that the 1st ext instruction was inputted, and the instruction decoding circuit 160 holds the immediate of 13 bits of the 1st ext instruction to ext1 register 172 (step 30).

[0089] When the condition of a state machine is S2, it is in (step 40) and the condition that the ext instruction was inputted continuously twice or more, and the instruction decoding circuit 160 holds the immediate of 13 bits of the inputted ext instruction to ext2 register 174 (step 50). For this reason, when an ext instruction continues 3 times or more, ext2 register will be overwritten by the immediate of the newest ext instruction. Therefore, when an ext instruction continues, the immediate of the first ext instruction and the last ext instruction becomes effective, and the middle ext instruction will be disregarded.

[0090] And since it is in the condition that the usual instruction was inputted when a condition is S0, when it judges whether the instruction code of this instruction contains an immediate (step 60) and an immediate is included, the immediate of this instruction code is outputted to an immediate generation circuit (step 70).

[0091] Next, the means which increases the number of bits of an immediate based on the held immediate is explained. As this means, the immediate generation circuit 170 has the signal generation means for immediate generation which is not illustrated, and generates the immediate extended based on the immediate generation signal generated with this signal generation means for immediate generation.

[0092] Drawing 5 is the timing-chart Fig. having shown the relation between the condition of the instruction code inputted and said state machine, and the signal for immediate generation which said signal generation circuit for immediate generation generates.

[0093] The signals which said signal generation circuit for immediate generation generates are the ext signal 530, and the ext_low signal 550 and the ext_high signal 560. 510 of drawing 5 expresses the instruction code inputted, and 540 expresses the condition of said state machine. Moreover, a clock signal 520 is a signal which is generated by the clock signal generator which is not illustrated in drawing 2, or is inputted from a clock input terminal. This clock signal 520 is used in order to take the synchronization of the various actuation in CPU. For example, synchronizing with the standup of this clock signal 520, the instruction address is outputted to the instruction address bus 120. Moreover, based on said instruction address, instruction code is read from ROM110 to every 1 of this clock signal 520 period (one machine cycle), and it is held at an instruction register 150. And the operation according to the read instruction code is completed within 1 machine cycle.

[0094] The ext signal 530 is a signal used as '1', when an ext instruction is inputted, and it is an input signal which produces the state transition of said state machine. That is, said signal generation circuit for immediate generation sets the ext signal 530 to '1', when the inputted instruction is an ext instruction, and when the inputted instruction is the usual instruction, it operates so that the ext signal 530 may be set to '0'.

[0095] The ext_low signal 550 is a signal outputted according to the condition of a state machine, and when a condition is 'S1', it is a delay signal delayed by one clock used as '1'. That is, when a condition is 'S1', said signal generation circuit for immediate generation takes a synchronization in the standup of the following clock signal, sets the ext_low signal 550 to '1', takes a synchronization in the standup of the following clock signal, and sets the ext_low signal 550 to '0'.

[0096] The ext_high signal 560 is a signal outputted according to the condition of a state machine, and when a condition is 'S2', it is a delay signal delayed by one clock used as '1'. That is, when a condition is 'S2', said signal generation circuit for immediate generation takes a synchronization in the standup of the

following clock signal 520, sets the ext_high signal 560 to '1', takes a synchronization in the standup of the following clock signal 520, and sets the ext_high signal 560 to '0'.

[0097] Drawing 6 is the flow chart Fig. having shown the algorithm with which an immediate generation circuit extends an immediate based on the condition of a state machine, and the signal for immediate generation, when the immediate is included in instruction code.

[0098] When there is no condition of a state machine S0, (step 110) and CPU are in the condition of processing the ext instruction which is a prefix instruction, and it is not in the condition of processing the usual instruction. In this case, the immediate generation circuit 170 does not generate an immediate (step 120). The prefix instruction itself is because activation of the operation in ALU in the instruction-execution section 60 etc. is not performed, so it is not necessary to generate an immediate at the time of processing of a prefix instruction (when the condition of a state machine is S1 or S2).

[0099] It is in the condition that the condition of a state machine is processing the usual instruction which is a target instruction after (step 110) and one ext instruction by S0 in ext_low signal 550='1'. In this case, the immediate of 19 bits which connects the immediate of 13 bits contained in the instruction code of an ext instruction of the point currently held at EXT1 register 172 and the immediate of 6 bits contained in the instruction code of this target instruction inputted through the signal line 176 for immediate generation, and can do it is zero-extended or sign extended at 32 bits (step 140).

[0100] It is in the condition that the condition of a state machine is processing the usual instruction which is a target instruction by S0 after (step 150) and the inputted ext instruction which carried out multiple-times continuation in ext_high signal 560='1'. In this case, the immediate of 13 bits contained in the instruction code of an ext instruction of the No. 1 beyond currently held at EXT1 register 172, The immediate of 13 bits contained in the instruction code of an ext instruction of the very end currently held at EXT2 register 174, The immediate of 6 bits contained in the instruction code of this target instruction inputted through the signal line 176 for immediate generation is connected, and the immediate of 32 bits is generated (step 160).

[0101] It is in the condition that the condition of a state machine is processing the usual instruction (the usual instruction whose last instruction is not an ext instruction) it is not a target instruction in [whose] ext_low signal 550='0' and ext_high signal 560='0' by S0. In this case, the immediate of 6 bits contained in the instruction code of this usual instruction inputted through the signal line 176 for immediate generation is zero-extended or sign extended at 32 bits (step 170).

[0102] Thus, the instruction decoding circuit 160 and the immediate generation circuit 170 function using the immediate of an ext instruction as an immediate escape means which increases the number of bits of the immediate of a target instruction of consecutiveness.

[0103] In addition, said processing is the case where the immediate is included in instruction code. However, even when the immediate is not included in instruction code, it is made not to use the immediate in instruction code by processing of step 140, and processing of step 160, and can apply like said processing by excluding processing of step 170.

[0104] By carrying out multiple-times activation of the ext instruction continuously, the contents of operation of an ext instruction of the 2nd henceforth are extensible. That is, the instruction decoding circuit 160, the immediate generation circuit 170, EXT1 register 172, and EXT2 register 174 function by taking a configuration which was mentioned above based on the count of a continuation input of a prefix instruction as a prefix instruction expansion means to extend and interpret the contents of operation of this prefix instruction.

[0105] 3. Explain taking the case of the case of an instruction (henceforth Type 1 instruction for convenience) of the type which specifies an immediate and a general-purpose register for the gestalt of typical implementation of the immediate escape by the ext instruction of example this invention of an immediate escape of the target instruction using an ext instruction (prefix instruction) in instruction code, and calculates this immediate and the value stored in the general-purpose register.

[0106] First, the configuration of the instruction code of Type 1 instruction and an ext instruction is

explained. Drawing 7 (A) is drawing having shown the bit field of the instruction code 210 of this Type 1 instruction, and drawing 7 (B) shows the bit field of the instruction code 220 of an ext instruction. The figure on a bit field shows the location of a bit, and as shown in this drawing (A) and (B), instruction code has the field of 16-bit width of face from the bit 15 to the bit 0.

[0107] As shown in this drawing (A), – DO 210 of instruction KO of Type 1 instruction has the 4-bit register appointed field 218 in the bit 0 from the bit 15 to the bit 13 from the field 214 which specifies the operation code of a triplet, the field 216 which specifies the immediate of 6 bits as a bit 4 from a bit 9, and the bit 3 from the bit 12 to the class appointed field 212 of a triplet, and the bit 10.

[0108] In the class appointed field 212, it is the field which defines the group of an instruction. Since the group division of the instruction used with this microcomputer is carried out according to the predetermined Ruhr, it is specified as the class appointed field which group it is the instruction belonging to.

[0109] The operation code which specifies the actuation function of an instruction is stored in the field 214 which specifies an operation code. In addition, since the operation code which belongs to the class for every class was decided, the operation code belonging to the class specified in said class appointed field 212 is stored in the field 214 which specifies this operation code.

[0110] Moreover, the value of the immediate (imm6) of 6 bits is stored in the field 216 which specifies said immediate, and the code which shows one of general-purpose registers (rd) is stored in the register appointed field 218.

[0111] Type 1 instruction is an instruction which performs the operation shown by said operation code to said immediate (imm6) and general-purpose register (rd), and performs actuation which writes in a result to a general-purpose register (rd).

[0112] Moreover, as shown in this drawing (B), the instruction code 220 of an ext instruction has the class appointed field 222 of a triplet, and the field 224 which specifies the immediate of 13 bits from a bit 12 at a bit 0 in the bit 13 from the bit 15. The value of the immediate (imm13) of 13 bits is stored in the field 224 which specifies said immediate.

[0113] Although this ext instruction does not have the operation code, the prefix instruction which the microcomputer of the gestalt of this operation processes is one kind of ext instruction, and since it is assigning the predetermined class to this prefix instruction, it can be distinguished by the class specified as the class appointed field 222 as it is an ext instruction.

[0114] Since an ext instruction is a prefix instruction, if independent, the operation in ALU in CPU etc. is not performed at all, but in case a target instruction of consecutiveness is executed, it has the function which extends the immediate used for activation of the target instruction. For example, when the instruction which contains the immediate in instruction code like Type 1 instruction turns into a target instruction of an ext instruction, it has the function which extends the immediate contained in the instruction code of Type 1 instruction using the immediate of 6 bits of an ext instruction (imm6) at the time of type 1 instruction execution:

[0115] Next, the escape of the immediate at the time of said Type 1 instruction being executed is explained in detail.

[0116] First, the contents of activation when Type 1 instruction is executed independently are explained. In drawing 2, Type 1 instruction is first inputted into an instruction register 150 through the instruction data bus 120 from ROM110. And it decodes with an instruction in the code circuit 160, and the operation according to the contents is performed. In Type 1 instruction, the data stored in the general-purpose register (rd) specified by instruction code are inputted into ALU190 through a data bus 182 from a register file 90. Moreover, the immediate (imm6) specified by instruction code is started by the instruction decoding circuit, and is inputted into the immediate generation circuit 170 through the signal line 176 for immediate generation. And said immediate (imm6) is zero-extended or sign extended at 32 bits, and said extended immediate is inputted into ALU190 in the immediate generation circuit 170. And ALU190 performs the operation shown by the operation code of Type 1 instruction, and stores the

result of an operation in the general-purpose register (rd) of a register file 90 through a data bus 192.

[0117] Said Type 1 instruction can also be executed independently and it is also possible to perform combining last 1 or two or more ext instructions. Although it is the same as that of the case of said independent activation fundamentally also about the contents of activation at the time of performing combining an ext instruction, the processes ***** (ed) in the immediate extended from the immediate (imm6) started by the instruction decoding circuit differ. That is, when it performs combining an ext instruction, the escape of imm6 is performed using the immediate of an ext instruction of the point currently held at EXT1 register 172 or EXT2 register 174. This process and the process in which the immediate of an ext instruction is held in advance of this process at EXT1 register 172 or EXT2 register 174 are as drawing 3 – drawing 6 having explained. Therefore, the extended immediates which are generated differ in the immediate generation circuit 170 by whether it performed combining 1 of whether Type 1 instruction was executed independently and a just before, or two or more ext instructions.

[0118] Drawing 8 (A) – (C) is the field Fig. of the formula showing the operation of each of said ****, and the extended immediate which is used for activation of this operation.

[0119] Drawing 8 (A) is the bit field Fig. of the formula showing operation when Type 1 instruction is executed independently, and the extended immediate 230 which is used for activation. As shown in this drawing, the immediate of 6 bits of Type 1 instruction (imm6) is extended by the approach of a zero escape or a sign escape either, and turns into the immediate 230 of 32 bits. When a zero escape is carried out, the field 232 of a bit 6 to the bit 31 serves as zero altogether, and when a sign escape is carried out, the field 232 of a bit 6 to the bit 31 serves as the most significant bit 5 of imm6, i.e., a bit, and the same bit altogether.

[0120] Drawing 8 (B) is the bit field Fig. of the formula showing the operation at the time of performing combining the ext instruction of last one, and the extended immediate 240 which is used for activation of this operation. As shown in this drawing, the immediate of 6 bits of Type 1 instruction (imm6) is set to the field 246 of a bit 5 to the bit 0, the immediate of 13 bits of the ext instruction of last one (imm13) is set to a bit 6 from a bit 18, and the immediate (imm19) of 19 bits is generated. And said immediate (imm19) of 19 bits turns into the immediate 240 of 32 bits by the extended approach of a zero escape or a sign escape either. When a zero escape is carried out, the 19 to 31 bits field 242 serves as zero altogether, and when a sign escape is carried out, the 19 to 31 bits field 242 becomes the most significant bit of imm19, i.e., the same bit as the 18th bit, altogether.

[0121] Drawing 8 (C) is the bit field Fig. of the formula showing the operation at the time of performing combining two ext instructions, and the extended immediate 250 which is used for activation of this operation. As shown in this drawing, the immediate of 6 bits of Type 1 instruction (imm6) is set to the field 256 of a bit 5 to the bit 0, the immediate of 13 bits of the 1st ext instruction (imm13) is set to a bit 19 from a bit 31, the immediate of 13 bits of the 2nd ext instruction (imm13) is set to a bit 6 from a bit 18, and the immediate (imm32) 250 of 32 bits is generated.

[0122] In addition, combining two ext instructions or more, the activation case is constituted so that the ext instruction of the beginning and the last may become effective. That is, the first ext instruction functions as said 1st ext instruction similarly, and the last ext instruction functions as said 2nd ext instruction similarly.

[0123] 4. Explain the gestalt of typical implementation of the contents escape of operation of the target instruction by the prefix instruction of example this invention of the contents escape of operation of the target instruction using a prefix instruction taking the case of the case of an instruction (henceforth Type 2 instruction for convenience) of the type which calculates using 2 general-purpose registers.

[0124] First, the configuration of the instruction code of Type 2 instruction is explained. Drawing 9 is drawing having shown the bit field of the instruction code 260 of this Type 2 instruction.

[0125] As shown in this drawing, the instruction code 260 of Type 2 instruction has the 4-bit register appointed field 268 in the bit 4 from the bit 15 to the bit 13 from the field 264 which specifies a 5-bit operation code, and the bit 7 from the bit 3 to the 4-bit register appointed field 266 and the bit 0 from

the bit 12 to the class appointed field 262 of a triplet, and the bit 8.

[0126] Moreover, the code which shows one of general-purpose registers (rs) and (rd), respectively is stored in said register appointed fields 266 and 268. the register source is meant in rs and a register DISUTE nation is meant in rd — it is. Type 2 instructions are 2 operand instructions which perform the operation shown so much in the data stored in the general-purpose register (rs), and the data stored in the general-purpose register (rd) by the operation code, and perform actuation which writes a result in a general-purpose register (rd).

[0127] Next, actuation of CPU when Type 2 instruction is executed independently is explained. In drawing 2, Type 2 instruction is first inputted into an instruction register 150 through the instruction data bus 130 from ROM110. And it decodes in the instruction decoding circuit 160, and the operation according to the contents is performed. In Type 2 instruction, the data stored in the general-purpose register (rd) specified by instruction code are inputted into ALU190 through a data bus 182 from a register file 90. Moreover, the data stored in the general-purpose register (rs) specified by instruction code are inputted into ALU190 through a data bus 184 from a register file 90. And ALU190 performs the operation shown by the operation code of Type 2 instruction, and stores the result of an operation in the general-purpose register (rd) of a register file 90 through a data bus 192.

[0128] Said Type 2 instruction can also be executed independently and it is also possible to perform combining last 1 or two or more ext instructions.

[0129] Drawing 10 (A) – (C) is drawing having shown the bit field of the formula which expressed the operation in the case of performing combining last 1 or two ext instructions when Type 2 instruction was executed independently, and the immediate used for activation.

[0130] Drawing 10 (A) is a formula showing operation when Type 2 instruction is executed independently. Since original operation of Type 2 instruction is performed as shown in this formula when performing independently, an immediate is not used. In addition, it is as having mentioned above about the contents of activation when Type 2 instruction is executed independently.

[0131] Drawing 10 (B) is the bit field Fig. of the formula showing operation when Type 2 instruction is executed combining the ext instruction of last one, and the extended immediate 270 which is used for activation of this operation. As shown in this formula, when Type 2 instruction is executed combining the ext instruction of last one, the contents of operation extend the result of an operation of the immediate (imm13) generated based on an ext instruction, and the contents of the general-purpose register (rs) so that it may write in a general-purpose register (rd). The immediate (imm13) contained in an ext instruction carries out a zero escape at 32 bits, and an immediate 270 is generated.

[0132] The contents of activation when Type 2 instruction is executed combining the ext instruction of 1 are explained. In drawing 2, Type 2 instruction is first inputted into an instruction register 150 through the instruction data bus 130 from ROM110. And although it decodes in the instruction decoding circuit 160, the following operation is performed when this instruction is a target instruction at this time (when the condition of said state machine is except S0). That is, the data stored in the general-purpose register (rs) specified by instruction code are inputted into ALU190 through a data bus 182 from a register file 90. Moreover, a zero escape is carried out by the immediate generation circuit 170, said immediate 270 is generated, and the immediate (imm13) of an ext instruction just before being held at EXT1 register 172 is inputted into ALU190. And ALU190 performs the operation shown by the operation code of Type 2 instruction, and stores the result of an operation in the general-purpose register (rd) of a register file 90 through a data bus 192.

[0133] In addition, the process in which the immediate of an ext instruction is held in advance of generation ***** and this process at EXT1 register 172 or EXT2 register 174 in the immediate used for activation of Type 2 instruction using the immediate of an ext instruction of the point currently held at EXT1 register 172 or EXT2 register 174 is as drawing 3 – drawing 6 having explained.

[0134] Therefore, when Type 2 instruction is executed combining an ext instruction, it has the effectiveness same with having executed 3 operand instructions which have an immediate and two

general-purpose registers in an operand.

[0135] Drawing 10 (C) is the bit field Fig. of the formula showing operation when Type 2 instruction is executed combining two ext instructions, and the extended immediate 280 which is used for activation of this operation. As shown in this formula, when Type 2 instruction is executed combining two ext instructions, the contents of operation extend the result of an operation of the immediate (imm26) generated based on two ext instructions, and the contents of the general-purpose register (rs) so that it may write in a general-purpose register (rd). An immediate 280 sets to a bit 13 the immediate imm13 contained in the 1st ext instruction from a bit 25, it is zero-extended or sign extended and 32 bits generates imm26 which set to the bit 0 the immediate contained in the 2nd ext instruction from the bit 12.

[0136] In addition, when it performs combining two ext instructions or more, it is constituted so that the ext instruction of the beginning and the last may become effective. That is, the first ext instruction functions as said 1st ext instruction similarly, and the last ext instruction functions as said 2nd ext instruction similarly.

[0137] the thing [process / a thing / in which the immediate generation circuit 170 generates said immediate 280] although the contents of activation when Type 2 instruction is executed combining the ext instruction of 2 times or multiple times are the same as that of the case where it performs combining one ext instruction, fundamentally — **** — it comes.

[0138] That is, the immediate generation circuit 170 generates said immediate 280 using the immediate (imm13) of the 1st ext instruction currently held at EXT1 register 172, and the immediate (imm13) of an ext instruction of the last currently held at EXT2 register 174.

[0139] In addition, the process in which the immediate of an ext instruction is held in advance of generation ***** and this process at EXT1 register 172 or EXT2 register 174 in the immediate used for activation of Type 2 instruction using the immediate of an ext instruction of the point currently held at EXT1 register 172 or EXT2 register 174 is as drawing 3 – drawing 6 having explained.

[0140] Thus, since the contents of operation of a target instruction are extensible by using an ext instruction, an instruction decoding circuit and an immediate generation circuit function as a target instruction expansion means.

[0141] Moreover, since the immediate which extended [zero-] or extended [sign-] the thing which is 13 bits of two ext instructions, and which was connected the immediate (imm13) can be used when Type 2 instruction is executed combining the ext instruction of 2 times or multiple times, the immediate which has the larger number of effective bits than the case where it combines with one ext instruction can be processed.

[0142] Thus, since a function is extended compared with the case where it performs combining one ext instruction when Type 2 instruction is executed combining the ext instruction of 2 times or multiple times, an instruction decoding circuit and an immediate generation circuit function as a prefix instruction expansion means.

[0143] 5. the example of an immediate escape of the address of the target instruction using a prefix instruction — explain the escape of the address at the time of the target instruction execution by the prefix instruction of this invention below. Drawing 11 illustrates a part required for explanation of the configuration of the semiconductor integrated circuit built in said microcomputer. It is the same as that of the configuration shown in drawing 2 fundamentally, and the same sign is attached about the part which has the same function as drawing 2 . In addition, since it is easy, the EXT1 register 172 grade of drawing 2 omits, and is illustrating only the instruction decoding circuit 160 and the immediate generation circuit 170. Moreover, in addition to the configuration of drawing 2 , RAM320, the program counter PC 330, the address-arithmetic machine 340, and the X bus 350 are illustrated. In addition, the program counter PC 330 is illustrated independently [a register file 90] on account of explanation. Moreover, although the address-arithmetic machine 340 calculates four operations etc., it takes up only the function which carries out an add operation for convenience, and this example explains it.

[0144] Said RAM320 has memorized the data used for activation. The address-arithmetic machine 340 calculates the address of the data memorized by said RAM320, and it is constituted so that the value stored in the immediate generated in the immediate generation circuit 170 and the value of a program counter PC 330, or the register file 90 may be inputted. The address of the instruction under current activation is stored in the program counter PC 330. And the address of RAM320 is specified through the data address bus 310 with the address calculated with said address-arithmetic vessel 340, or the address of an instruction of ROM110 is specified through the instruction address bus 120. And through the data bus 140 or the instruction data bus 130, the data or instruction code of the address specified with said each address buses 310 and 120 is inputted into CPU30, and the way configuration is carried out.

[0145] Next, it has the immediate of the address in an operand and the escape of the immediate of the address by the prefix instruction in an instruction (henceforth Type 3 instruction for convenience) of the type which uses for activation the relative address which added this immediate to the value stored in the program counter PC 330 is explained.

[0146] Drawing 12 is drawing having shown the bit field of the instruction code 290 of PC relative subroutine call instruction (instruction which calls a subroutine with a relative address) which is an example of Type 3 instruction.

[0147] As shown in this drawing, the instruction code 290 of Type 3 instruction has the field 298 which specifies the immediate of 8 bits at a bit 0 from the bit 15 to the bit 13 from the field 294 which specifies a 4-bit operation code, the d bit field 296 which specifies the bit accompanying the predetermined operation code of a bit 8, and the bit 7 from the bit 12 to the class appointed field 292 of a triplet, and the bit 9.

[0148] It is the instruction which branches to this address by making into the address the value adding the immediate of 32 bits obtained by making PC relative subroutine call instruction into the value (sign9) of 9 bits in shifting to the 1-bit left the immediate (imm8) of 8 bits stored in the field 298 which specifies said immediate, and carrying out the sign escape of this value of sign9, and the value of PC (program counter).

[0149] It is with drawing 11 and actuation of CPU in case this PC relative subroutine call instruction is executed independently is explained. It is shifted to the 1-bit left by the immediate generation circuit 170, a sign escape is carried out, and the immediate of 8 bit of these ***** rare ** turns into an immediate (sign9) of 32 bits by it. and — this — the immediate (sign9) of 32 bits is inputted into the address-arithmetic machine 340. Moreover, the value of the address stored in the program counter PC 330 is also inputted into the address-arithmetic machine 340. The address-arithmetic machine 340 adds said immediate (sign9) and the value of the address stored in the program counter PC 330, and generates a branch address. And the subroutine stored in the branch address will be performed.

[0150] A Type 3 instruction like PC relative subroutine call instruction can also be executed independently, and it is also possible to perform combining last 1 or two or more ext instructions. Although it is the same as that of the case of said independent activation fundamentally also about the contents of activation at the time of performing combining an ext instruction, it is the immediate generation circuit 170 of the instruction decoding circuit 160, and the processes in which the immediate extended from the immediate (imm8) started from instruction code is generated differ. That is, when it performs combining an ext instruction, the escape of imm8 is performed using the immediate of an ext instruction of the point currently held at EXT1 register 172 or EXT2 register 174. This process and the process in which the immediate of an ext instruction is held in advance of this process at EXT1 register 172 or EXT2 register 174 are as drawing 3 – drawing 6 having explained. Therefore, the address of a branching place can be changed by whether it performed combining 1 of whether Type 3 instruction was executed independently and a just before, or two or more ext instructions.

[0151] Drawing 13 (A) – (C) is the field Fig. of the formula showing the operation of Type 3 instruction of each of said ****, and the extended immediate which is used for activation of this operation.

[0152] Drawing 13 (A) is the bit field Fig. of the formula showing operation when Type 3 instruction is executed independently, and the extended immediate 400 which is used for activation. Since 16 bit fixed length's instruction code is used for CPU30, the value of the instruction address always serves as an even number byte address, consequently the least significant bit of the instruction address is always set to "0." Therefore, as shown in this drawing in the case of an escape, the immediate of 8 bits of Type 3 instruction (imm8) turns into an immediate of 9 bit of shift ** on the 1-bit left. and — this — a sign escape is carried out and the immediate of 9 bits turns into the immediate 400 (sign9) of 32 bits.

[0153] Drawing 13 (B) is the bit field Fig. of the formula showing operation when Type 3 instruction is executed combining the ext instruction of last one, and the extended immediate 410 which is used for activation of this operation. As shown in this drawing, the immediate of 21 bits which combined the immediate of 8 bits of Type 3 instruction (imm8) and the immediate of 13 bits of an ext instruction (imm13) is shifted to the 1-bit left, and turns into an immediate of 22 bits. this — a sign escape is carried out and the immediate of 22 bits turns into the immediate 410 (sign22) of 32 bits.

[0154] Drawing 13 (C) is the bit field Fig. of the formula showing operation when Type 3 instruction is executed combining two ext instructions, and the extended immediate 420 which is used for activation of this operation. As shown in this drawing, the immediate of 31 bits which combined the immediate of 8 bits of Type 3 instruction (imm8), the immediate of 10 bits of the 1st ext instruction (imm10), and the immediate of 13 bits of the 2nd ext instruction (imm13) is shifted to the 1-bit left, and turns into an immediate (sign32) of 32 bits.

[0155] In addition, combining two ext instructions or more, the activation case is constituted so that the ext instruction of the beginning and the last may become effective. That is, the first ext instruction functions as said 1st ext instruction similarly, and the last ext instruction functions as said 2nd ext instruction similarly.

[0156] 6. Explain the example which adds the displacement of the address using a prefix instruction to the example which adds displacement during an instruction of a target instruction using a prefix instruction, next an operand in an instruction (henceforth Type 4 instruction for convenience) of the type which performs using as the address the value in which the immediate was stored by owner **** and the operand at the specified register.

[0157] The configuration of instruction code is common in that it has in an operand the appointed field of Type 2 instruction shown in drawing 9, and two registers. Although the contents of the operation code differ, since there is especially no effect in explanation of the gestalt of this operation, explanation is omitted. The load instruction which is an example of Type 4 instruction is taken and explained to an example. A load instruction is an instruction for loading to the register which shows the data memorized to the address specified by rs to rd.

[0158] First, actuation of CPU in case this load instruction is executed independently is explained. CPU reads the data which make the address the value stored in the general-purpose register (rs) specified by instruction code from RAM320, and stores them in the general-purpose register (rd) specified by instruction code.

[0159] A Type 4 instruction like a load instruction can also be executed independently, and it is also possible to perform combining last 1 or two or more ext instructions. When it performs combining an ext instruction, displacement is added by the immediate contained in the instruction code of an ext instruction, and Type 4 instruction is executed.

[0160] Drawing 14 (A) and (B) are drawings having shown the displacement generated when executing Type 4 instruction combining last 1 or two ext instructions.

[0161] Drawing 14 (A) should express up displacement when it constructs with one ext instruction, it is united and Type 4 instruction is executed, and requires it. As shown in this drawing, the zero escape of the immediate (imm13) of an ext instruction of last one is carried out, and the 32-bit immediate (imm13) 430 is generated.

[0162] The contents of activation when Type 4 instruction is executed combining the ext instruction of

1 are explained using drawing 11 . First, Type 4 instruction is inputted into an instruction register 150 through the instruction data bus 130 from ROM110. And although it decodes in the instruction decoding circuit 160, when this instruction is a target instruction at this time, the address stored in the general-purpose register (rs) specified by instruction code is taken out from a register file 90, and is inputted into the address-arithmetic machine 340 through the X bus 350 (when the condition of said state machine is except S0).

[0163] Moreover, a zero escape is carried out by the immediate generation circuit 170, the immediate 430 of 32 bits is generated, and the immediate (imm13) of an ext instruction just before being held at EXT1 register 172 is inputted into the address-arithmetic machine 340. The address-arithmetic machine 340 adds these two inputs, and generates the address. The address of RAM320 is specified through the data address bus 310 with the generated address, the data stored in this address are inputted into a register file 90 through a data bus 140, and it stores in the general-purpose register (rd) specified by instruction code.

[0164] Drawing 14 (B) expresses the displacement at the time of the group bubble of the Type 4 instruction being carried out to two ext instructions, and performing. As shown in this drawing, the immediate (imm13) of the 1st ext instruction is set to the 13-bit field 444 of a bit 25 to the bit 13, the immediate (imm13) of the 2nd ext instruction is set to the 13-bit field 446 of a bit 12 to the bit 0, a zero escape is carried out, and the immediate 440 (imm26) of 32 bits is generated.

[0165] The contents of activation when Type 4 instruction is executed combining two ext instructions are explained using drawing 11 . Although it is fundamentally [as the case where it combines with one ext instruction also in this case] the same, the immediates generated as displacement differ. When it combines with two ext instructions, the immediate 440 as shown in drawing 14 (B) is generated using the immediate of two ext instructions currently held at EXT1 register 172 and EXT2 register 174, and this immediate 440 is inputted into the address-arithmetic machine 340 as displacement.

[0166] In addition, the process in which the immediate of an ext instruction is held in advance of generation ***** and this process at EXT1 register 172 or EXT2 register 174 in the immediate used for activation of Type 4 instruction using the immediate of an ext instruction of the point currently held at EXT1 register 172 or EXT2 register 174 is as drawing 3 - drawing 6 having explained.

[0167] Therefore, when Type 4 instruction is executed combining an ext instruction, the contents of operation are extended so that displacement may be added and performed to an operand.

[0168] (Gestalt 2 of operation) Drawing 15 is the hardware block diagram of the microcomputer of the gestalt of this operation.

[0169] This microcomputer 10 is a 32-bit microcontroller. CPU30, ROM110, RAM320 and the RF dispatch circuit 710, the low frequency dispatch circuit 720, a reset circuit 730, the 750 or 8 bit programmable timer 760 of 740 or 16 bit programmable timers of prescalers, the clock timer 770, Intelligent DMA780, a high speed DMA 790, the interruption controller 600, serial interface 610, a bus control unit (BCU) 620, A/D converter 630, D/A converter 640, input port 660, an output port 660, The various bus 680 grades and the various pin 690 grades which connect I/O Port 670 and them are included.

[0170] Said CPU30 processes 16-bit fixed-length instruction code, and the data size at the time of activation is 32-bit CPU. This CPU30 has the configuration of the gestalt 1 of operation mentioned above, and functions as an instruction code analysis means, an immediate escape means, a target instruction expansion means, a prefix instruction expansion means, and an instruction-execution means.

[0171] Therefore, since the handling of a big immediate and 3 operand instructions which cannot be described by short instruction code can be executed even if it uses 16-bit instruction code, a microcomputer with the sufficient utilization ratio of memory can be offered. Moreover, if the number of instructions made to memorize is the same when it constitutes this microcomputer as a semiconductor integrated circuit, compared with the case where the fixed-length instruction code of 32-bit width of face is used, capacity is good in one half. Therefore, size of a chip can be made small and the good

semiconductor integrated circuit equipment of the yield can be manufactured.

[0172] The microcomputer of this invention is applicable to personal computer peripheral devices, such as a printer, and various kinds of electronic equipment, such as a pocket device. If it is made this appearance, since the good information processing circuit of the utilization ratio of memory can be built in with an easy configuration, cheap and highly efficient electronic equipment can be offered.

[0173] In addition, what [not only] was explained in the above-mentioned example but various deformation implementation is possible for this invention.

[0174] For example, in the above-mentioned example, although it explained taking the case of the case where 16-bit fixed-length instruction code is used, in CPU or the microcomputer which can process an effective example, i.e., 32-bit data, especially when raising the utilization ratio of memory, it is not restricted to this. It cannot be based on the number of bits which can be processed with CPU or a microcomputer, and the number of bits of instruction code, but this invention can be applied. Moreover, it cannot be because instruction code is a fixed length, but this invention can be applied.

[0175] And CPU or the microcomputer which the escape of the immediate of instruction code is performed and can make a change of the contents of operation easily with an easy configuration can be offered by applying this invention.

[0176] Moreover, when continuation activation of the ext instruction was carried out twice or more, the case where only the ext instruction of the 1st last functioned effectively was explained, but you may constitute from a gestalt of this above-mentioned implementation so that three ext instructions or more may be confirmed. Moreover, selection of the ext instruction to confirm is not restricted to the 1st last, but its ** useless **** is good according to the predetermined Ruhr.

[0177] Moreover, the example to which an ext instruction function is changed is not restricted to the above-mentioned example by combining two or more ext instructions, either.

[0178] Moreover, although the gestalt of this above-mentioned implementation explained the case where a prefix instruction was one kind of ext instruction, the prefix instruction of a class with which plurality differs may be set up.

[0179]

[Translation done.]

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the functional block diagram of CPU of the gestalt of operation of this invention.

[Drawing 2] A part required for explanation of the configuration of the semiconductor integrated circuit built in the microcomputer is illustrated.

[Drawing 3] It is drawing showing transition of the condition of the state machine which an instruction decoding circuit has.

[Drawing 4] It is the flow chart Fig. having shown the algorithm of the actuation which an instruction decoding circuit performs about maintenance of an immediate.

[Drawing 5] It is the timing-chart Fig. having shown the relation of the instruction code inputted and said state machine, and the signal for immediate generation which said signal generation circuit for immediate generation generates.

[Drawing 6] An immediate generation circuit is the flow chart Fig. having shown the algorithm which extends an immediate based on the condition of a state machine, and the signal for immediate generation.

[Drawing 7] This drawing (A) and (B) are drawings having shown the bit field of the instruction code of Type 1 instruction and an ext instruction.

[Drawing 8] This drawing (A) – (C) is the field Fig. of the extended immediate which is used for activation of the formula and operation of having expressed operation.

[Drawing 9] It is drawing having shown the bit field of the instruction code of Type 2 instruction.

[Drawing 10] This drawing (A) – (C) is drawing having shown the bit field of the formula showing the operation of Type 2 instruction, and the immediate used for activation.

[Drawing 11] A part required for explanation of the configuration of the semiconductor integrated circuit built in the microcomputer is illustrated.

[Drawing 12] It is drawing having shown the bit field of the instruction code of Type 3 instruction.

[Drawing 13] This drawing (A) – (C) is the field Fig. of the formula showing the operation of Type 3 instruction, and the extended immediate which is used for activation of this operation.

[Drawing 14] This drawing (A) and (B) are drawings having shown the displacement generated when executing Type 4 instruction.

[Drawing 15] It is the hardware block diagram of the microcomputer of the gestalt 2 of operation.

[Description of Notations]

10 Microcomputer

30 CPU

40 Instruction Decoding Section

50 Ext Instruction-Processing Section

60 Instruction-Execution Section

70 Data Operation Part

80 Address-Arithmetic Section

90 Register File

110 ROM

120 Instruction Address Bus

130 Instruction Data Bus

140 Data Bus

150 Instruction Register

160 Instruction Decoding Circuit

161 Control Signal

170 Immediate Generation Circuit

172 EXT1 Register

174 EXT2 Register

190 ALU

310 Data Address Bus

320 RAM

330 Program Counter PC

340 Address-Arithmetic Machine

[Translation done.]

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-231070

(43)公開日 平成9年(1997)9月5日

(51)Int.Cl. ⁸	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/30	3 1 0		G 0 6 F 9/30	3 1 0 E
	3 5 0			3 5 0 A
9/305				3 4 0 F

審査請求 未請求 請求項の数20 F D (全 21 頁)

(21)出願番号 特願平8-353167

(22)出願日 平成8年(1996)12月16日

(31)優先権主張番号 特願平7-332207

(32)優先日 平7(1995)12月20日

(33)優先権主張国 日本 (J P)

(71)出願人 000002369

セイコーエプソン株式会社

東京都新宿区西新宿2丁目4番1号

(72)発明者 久保田 哲

長野県諏訪市大和3丁目3番5号 セイコ

ーエプソン株式会社内

(72)発明者 工藤 真

長野県諏訪市大和3丁目3番5号 セイコ

ーエプソン株式会社内

(72)発明者 宮山 芳幸

長野県諏訪市大和3丁目3番5号 セイコ

ーエプソン株式会社内

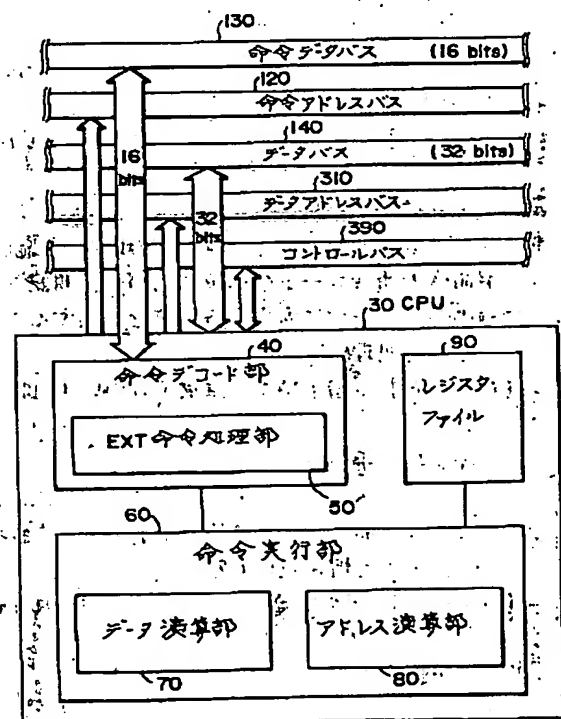
(74)代理人 弁理士 井上 一 (外2名)

(54)【発明の名称】 情報処理回路、半導体集積回路装置、マイクロコンピュータ、電子機器

(57)【要約】

【課題】必要に応じて命令コードに含まれる即値を拡張する機能を有する情報処理回路、半導体集積回路装置、マイクロコンピュータ、電子機器を提供する。

【解決手段】所与のターゲット命令と、該ターゲット命令の機能を拡張するためのプリフィックス命令が入力される情報処理回路である。該情報処理回路は、入力した命令コードを解釈し、該命令を実行するために必要な処理を行うのもで、前記命令デコード部40と、レジスタファイル90と、前記命令デコード部40が解析した命令のオペレーション内容に基づき該命令の実行をおこなう命令実行部60を含む。前記命令デコード部40は、プリフィックス命令により即値を拡張する処理を行うext命令処理部50を含む。



(2)

【特許請求の範囲】

【請求項1】 所与のターゲット命令と、該ターゲット命令の機能を拡張するためのプリフィックス命令が入力される情報処理回路であって、
前記ターゲット命令及びプリフィックス命令を入力し、該命令コードのオペレーション内容を解析する命令コード解析手段と、

前記命令コード解析手段が解析したオペレーション内容に基づき命令を実行する命令実行手段とを含み、

前記命令コード解析手段は、

前記プリフィックス命令に基づき、該プリフィックス命令の機能拡張の対象となるターゲット命令の実行に必要な即値を拡張する即値拡張手段を含むよう構成され、

前記命令実行手段は、

前記即値拡張手段により拡張された即値に基づきターゲット命令を実行することを特徴とする情報処理回路。

【請求項2】 請求項1において、

固定長の命令コードを入力して実行するよう構成されていることを特徴とする情報処理回路。

【請求項3】 請求項2において、

前記固定長命令コードのビット幅が、前記情報処理回路が処理出来るデータ又はアドレスのビット幅以下であることを特徴とする情報処理回路。

【請求項4】 請求項3において前記情報処理回路は、16のビット幅の固定長の命令コードを入力するよう構成され、

前記即値拡張手段は、

前記プリフィックス命令に基づき、前記ターゲット命令の実行に必要な即値を32ビットの幅に拡張し、

前記命令実行手段は、

前記即値拡張手段で拡張された32ビットの即値を用いてターゲット命令を実行する算術論理演算手段を含むことを特徴とする情報処理回路。

【請求項5】 請求項1～請求項4のいずれかにおいて、

前記即値拡張手段は、

プリフィックス命令の命令コードの即値及び前記ターゲット命令の命令コードの即値に基づき、前記ターゲット命令の命令コードに含まれる即値を拡張することを特徴とする情報処理回路。

【請求項6】 請求項1～請求項5のいずれかにおいて、

前記即値拡張手段は、

プリフィックス命令の命令コードに含まれた即値を用いてターゲット命令の実行に使用するデータを拡張することを特徴とする情報処理回路。

【請求項7】 請求項1～請求項6のいずれかにおいて、

前記即値拡張手段は、

プリフィックス命令の命令コードに含まれた即値を用い

2

てターゲット命令の実行に使用するアドレスを拡張することを特徴とする情報処理回路。

【請求項8】 請求項1～請求項7のいずれかにおいて、

前記即値拡張手段は、

複数のプリフィックス命令の命令コードに含まれた各即値を用いて、該プリフィックス命令の機能拡張の対象となるターゲット命令の実行に必要な即値を拡張することを特徴とする情報処理回路。

10 【請求項9】 請求項8において、

前記複数のプリフィックス命令の各命令コードは、プリフィックス命令であることを判別するための共通のコードをそれぞれ含むことを特徴とする情報処理回路。

【請求項10】 請求項8又は請求項9のいずれかにおいて、

前記即値拡張手段は、

プリフィックス命令の連続入力回数に基づき決定される処理状態を記憶する処理状態記憶手段と、

前記処理状態記憶手段に記憶された処理状態に基づき、

20 入力されたプリフィックス命令の即値を第1のレジスタ～第mのレジスタ（mは1以上の整数）に保持する前記データ保持手段と、

前記データ保持手段の第1のレジスタ～第mのレジスタ（mは1以上の整数）に保持された即値に基づき、前記ターゲット命令の命令の実行に必要な即値を拡張して生成する手段とを含むことを特徴とする情報処理回路。

【請求項11】 請求項1～請求項10のいずれかにおいて、

前記命令コード解析手段は、

30 プリフィックス命令入力後にターゲット命令を入力した場合、該ターゲット命令のオペレーション内容を拡張して解釈するターゲット命令機能拡張手段を含み、

前記命令実行手段は、

前記ターゲット命令機能拡張手段で拡張されたオペレーション内容で該ターゲット命令を実行することを特徴とする情報処理回路。

【請求項12】 請求項11において、

前記ターゲット命令機能拡張手段は、

40 プリフィックス命令入力後に2オペランド命令である所定のターゲット命令を入力した場合、該ターゲット命令の2つのオペランドとプリフィックス命令に含まれた即値を用いて、該ターゲット命令のオペレーション内容を3オペランド命令に拡張して解釈することを特徴とする情報処理回路。

【請求項13】 請求項11において、

前記ターゲット命令機能拡張手段は、

プリフィックス命令入力後にレジスタ指定値を有する所定のターゲット命令を入力した場合、プリフィックス命令の命令コードに含まれた即値に基づき、ターゲット命令の命令コードに指定されたレジスタに格納されたデー

50

(3)

3
タ又はアドレスを修正して実行するようにオペレーション内容を拡張することを特徴とする情報処理回路。

【請求項14】 請求項13において、
前記ターゲット命令機能拡張手段は、
プリフィックス命令入力後にレジスタ指定値を有する所定のターゲット命令を入力した場合、プリフィックス命令の命令コードに含まれた即値に基づき、ターゲット命令の命令コードに指定されたレジスタに格納されたアドレスのディスプレースメントを作成し、該ディスプレースメントを用いて実行するようにオペレーション内容を拡張することを特徴とする情報処理回路。

【請求項15】 請求項1～請求項14のいずれかにおいて、
前記命令コード解析手段は、
プリフィックス命令の連続入力回数に基づき、プリフィックス命令のオペレーション内容を拡張するプリフィックス命令機能拡張手段を含むことを特徴とする情報処理回路。

【請求項16】 請求項1～請求項15のいずれかにおいて、
前記命令コード解析手段に入力する命令コードを記憶する命令コード記憶手段さらに含むことを特徴とする情報処理回路。

【請求項17】 請求項1～請求項16記載の情報処理回路を含むことを特徴とする半導体集積回路装置。

【請求項18】 請求項1～請求項17記載の情報処理回路を含むことを特徴とするマイクロコンピュータ。

【請求項19】 請求項18において、RISC方式であることを特徴とするマイクロコンピュータ。

【請求項20】 請求項18又は請求項19記載のマイクロコンピュータを用いて制御することを特徴とする電子機器。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、情報処理回路及び半導体集積回路装置、前記情報処理回路を内蔵するマイクロコンピュータ、該マイクロコンピュータを用いて構成された電子機器に関する。

【0002】

【背景の技術】従来、32ビットのデータを処理できるRISC方式のマイクロコンピュータでは、32ビット幅に固定された命令コードを用いられていた。その理由は、固定長ビット幅の命令コードを用いると、可変長ビット幅の命令コードを用いる場合に比べ、命令のデコードに要する時間を短縮でき、また、マイクロコンピュータの回路規模を小さくすることが出来るからである。

【0003】ところが、32ビットのマイクロコンピュータにおいても、特に32ビットも必要としない命令コードも多い。従って全ての命令の命令コードを32ビットで記述すると、命令コードに冗長な部分が生じる命令

4
が多くなり、メモリーの使用効率が悪くなる。

【0004】このような場合、冗長な命令コードを論理圧縮しておき、マイクロコンピュータの内部で元の命令にデコードしながら、命令を実行することも可能である。しかし、このような方式では、制御回路が複雑になるという問題があった。

【0005】

【発明が解決しようとする課題】そこで、本願の発明者は、制御回路を複雑にすることなくメモリーの使用効率を向上させるために、処理出来るデータのビット幅より短いビット幅の固定長命令コードを処理するマイクロコンピュータについての検討を行っていた。

【0006】しかし、例えば32ビット固定長の命令コードを単に16ビットの固定長にすると以下のような問題点が生じる。

【0007】通常命令コードにはオペコードや即値以外のオペランドも記述するため、16ビットの命令コードを使用しても、使用できる即値のビット数は16ビットよりもさらに小さいものとなる。すなわち、32ビットのデータを処理できるにもかかわらず、16ビットより小さい即値しか命令コードで指定できないという問題が生じる。

【0008】また、例えば、16ビットの命令コードでは、3オペランド命令のオペランドのフィールドを確保することは難しい。従って、この様に短い命令コードでは記述が難しいオペレーションをどの様に処理するのかという問題が生じる。

【0009】上記問題を解決するためには、短い命令コードに含まれる即値を、必要に応じて拡張する機能が必要となる。また、短い命令コードで記述出来ないオペレーションを実行するために、オペレーション内容を拡張する機能が必要となる。

【0010】本発明の目的は、制御回路を複雑にすることなく、必要に応じて命令コードに含まれる即値を拡張する機能を有する情報処理回路、半導体集積回路装置、マイクロコンピュータ、電子機器を提供することである。

【0011】また、本発明の他の目的は、制御回路を複雑にすることなく、オペレーション内容を拡張する機能を有する情報処理回路、半導体集積回路装置、マイクロコンピュータ、電子機器を提供することである。

【0012】

【課題を解決するための手段】前記目的を達成するため、請求項1の発明は、所与のターゲット命令と、該ターゲット命令の機能を拡張するためのプリフィックス命令が入力される情報処理回路であって、前記ターゲット命令及びプリフィックス命令を入力し、該命令コードのオペレーション内容を解析する命令コード解析手段と、前記命令コード解析手段が解析したオペレーション内容に基づき命令を実行する命令実行手段とを含み、前記命

(4)

5

令コード解析手段は、前記プリフィックス命令に基づき、該プリフィックス命令の機能拡張の対象となるターゲット命令の実行に必要な即値を拡張する即値拡張手段を含むよう構成され、前記命令実行手段は、前記即値拡張手段により拡張された即値に基づきターゲット命令を実行することを特徴とする。

【0013】命令コード解析手段は、入力した命令コードを解釈し、命令の対象となるデータの記憶されている記憶手段の番地を計算する等、命令実行手段が命令を実行するために必要な処理を行う。

【0014】本発明の情報処理回路には、ターゲット命令及び該ターゲット命令の機能を拡張するプリフィックス命令が入力される。プリフィックス命令は、それ単独では命令実行手段において実行されず、後続のターゲット命令が実行される際に、そのターゲット命令の機能を拡張する機能を有する。

【0015】即値拡張手段は、プリフィックス命令に基づき、ターゲット命令の実行に必要な即値を拡張する。ターゲット命令の実行に必要な即値とは、例えばターゲット命令の命令コードに含まれている即値や、プリフィックス命令の命令コードに含まれている即値であって、実行する際に必要となる即値をいう。

【0016】ここでいう即値の拡張とは、例えば通常のゼロ拡張やサイン拡張のほか、命令コード中に記載されている即値のビット幅を伸張して、伸張された部分に、所与のビットを補う場合も含む。すなわち、本発明によれば、プリフィックス命令に基づき、前記所与のビットを補って、即値を拡張することが出来る。

【0017】従って、命令コードのビット数を大きくする原因となる即値を含む命令コードのビット数を短く設定することが可能となる。このため、可変長命令コードをデコードするための複雑な制御回路を有しなくても、短い固定長命令コードの採用するという簡単な構成で、メモリーの使用効率を改善することができる。

【0018】請求項2の発明は、請求項1において、固定長の命令コードを入力して実行するよう構成されていることを特徴とする。

【0019】本発明によれば、固定長命令コードを使用することにより、可変長ビット幅の命令コードを用いる場合に比べ、命令のデコードに要する時間を短縮でき、また、情報処理回路の回路規模を小さくすることが出来る。

【0020】また、プリフィックス命令により、即値を拡張することが出来るため、命令コードを長くする原因となる即値を短く設定することができる。従って、固定長の命令コードの長さを短く設定することができるため、命令コードの冗長な部分を効率よく削減し、メモリー、特に命令コードを記憶するメモリーの使用効率を改善することができる。

【0021】請求項3の発明は、請求項2において、前

6

記固定長命令コードのビット幅が、前記情報処理回路が処理出来るデータ又はアドレスのビット幅以下であることを特徴とする。

【0022】本発明によれば、命令コードのビット幅を情報処理回路が処理できるデータ又はアドレスのビット幅以下の小さなビット幅にしても、前記情報処理回路が処理できるビット幅に即値を拡張をすることができる。この様に、命令コードのビット幅を情報処理回路が処理できるデータ又はアドレスのビット幅以下にできるため、メモリーの使用効率を改善することができる。

【0023】請求項4の発明は、請求項3において前記情報処理回路は、16のビット幅の固定長の命令コードを入力するよう構成され、前記即値拡張手段は、前記プリフィックス命令に基づき、前記ターゲット命令の実行に必要な即値を32ビットの幅に拡張し、前記命令実行手段は、前記即値拡張手段で拡張された32ビットの即値を用いてターゲット命令を実行する算術論理演算手段を含むことを特徴とする。

【0024】従来、32ビットのデータを処理できるRISC方式のマイクロコンピュータでは、固定長32ビットの命令コードを用いて、処理が行なわれていた。しかし、前記命令コードの中には、特に32ビットも必要としない命令コードも多く、これがメモリーの使用効率悪化の原因となっていた。

【0025】すなわち、通常、命令コードは該命令の基本的なオペレーションを定めるクラスコード、オペコードや、即値以外のオペランドで構成される。ここにおいて即値以外のオペランドとは、例えばソースレジスタ(r's)やデスティネーションレジスタ(r'd)に対応するコード等をさす。このため、16ビットの命令コードを使用する場合、使用できる即値のビット数はさらに小さいものとなり、通常6ビット程度となる。一般に、命令コードが短い程、即値のビット幅の確保が困難になる。しかし、本発明によれば、即値の拡張が行えるため、32ビットのデータを処理出来る情報処理回路においても、16ビットという短い命令コードが使用できる。このため、メモリーの使用効率大幅にを改善することができる。

【0026】請求項5の発明は、請求項1～請求項4のいずれかにおいて、前記即値拡張手段は、プリフィックス命令の命令コードの即値及び前記ターゲット命令の命令コードの即値に基づき、前記ターゲット命令の命令コードに含まれる即値を拡張することを特徴とする。

【0027】この様にすると、ターゲット命令の命令コード中含まれている即値を拡張して実行したい場合、ターゲット命令の実行に必要な即値の一部をターゲット命令の命令コードの即値として指定し、ターゲット命令の実行に必要な即値の残りの部分をプリフィックス命令の命令コードの即値として指定するという簡単な構成で、即値を拡張することが出来る。

(5)

7

【0028】従って、命令コード中の即値用のビット数を短かく設定することが可能となり、メモリの使用効率を改善することができる。

【0029】請求項6の発明は、請求項1～請求項5のいずれかにおいて、前記即値拡張手段は、プリフィックス命令の命令コードに含まれた即値を用いてターゲット命令の実行に使用するデータを拡張することを特徴とする。

【0030】ここにおいて、データとはアドレスに対比される概念であり、その値の内容がアドレス以外を表すものをさす。ターゲット命令の実行に使用するデータとは、ターゲット命令の命令コードに含まれている即値のデータや、ターゲット命令の命令コードには含まれていないが、実行する際に使用する即値のデータをいう。この様にすると、簡単な構成で拡張されたデータの即値を用いてターゲット命令を実行することが出来る。

【0031】請求項7の発明は、請求項1～請求項6のいずれかにおいて、前記即値拡張手段は、プリフィックス命令の命令コードに含まれた即値を用いてターゲット命令の実行に使用するアドレスを拡張することを特徴とする。

【0032】ここにおいて、アドレスとはデータに対比される概念であり、その値の内容が記憶装置内のある特定の位置や場所を示すための情報をいう。ターゲット命令の実行に使用するアドレスとは、ターゲット命令の命令コードに含まれている即値のアドレスや、ターゲット命令の命令コードには含まれていないが、実行する際に使用する即値のアドレスをいう。この様にすると、簡単な構成で拡張されたアドレスの即値を用いてターゲット命令を実行することが出来る。

【0033】請求項8の発明は、請求項1～請求項7のいずれかにおいて、前記即値拡張手段は、複数のプリフィックス命令の命令コードに含まれた各即値を用いて、該プリフィックス命令の機能拡張の対象となるターゲット命令の実行に必要な即値を拡張することを特徴とする。

【0034】この様にすると、複数のプリフィックス命令の即値を組み合わせることによって、ターゲット命令の実行に必要な即値を任意に拡張することが出来る。従って、短い命令コードを用いても、大きな即値を用いた実行が扱えるようになるため、命令コードの短縮による弊害を防ぐことが出来る。

【0035】請求項9の発明は、請求項8において、前記複数のプリフィックス命令の各命令コードは、プリフィックス命令であることを判別するための共通のコードをそれぞれ含むことを特徴とする。

【0036】本発明によれば、異なるプリフィックス命令を複数設ける必要が無い。このため、ビット数の少ない命令コードを処理する情報処理回路においても、他の種類の命令を割愛することなく、即値の拡張が可能とな

8

る。

【0037】請求項10の発明は、請求項8又は請求項9のいずれかにおいて、前記即値拡張手段は、プリフィックス命令の連続入力回数に基づき決定される処理状態を記憶する処理状態記憶手段と、前記処理状態記憶手段に記憶された処理状態に基づき、入力されたプリフィックス命令の即値を第1のレジスタ～第mのレジスタ（mは1以上の整数）に保持する前記データ保持手段と、前記データ保持手段の第1のレジスタ～第mのレジスタ

10 （mは1以上の整数）に保持された即値に基づき、前記ターゲット命令の命令の実行に必要な即値を拡張して生成する手段とを含むことを特徴とする。

【0038】ここにおいてプリフィックス命令の連続入力回数に基づき決定される処理状態とは、連続入力回数そのものによって決定される処理状態でもよいし、連続入力回数からある特定のルールに従って一意的に導き出される処理状態でもよい。例えば、プリフィックス命令を3回以上くりかえして入力した場合、3回目以降は常に同じ処理状態となるよう構成してもよい。そして、この様に決定された処理状態を参照してプリフィックス命令の即値を即値保持手段に保持し、該保持された即値を組み合わせることによって即値を拡張することが出来る。この様にすると、簡単な構成で、プリフィックス命令の連続した入力回数に基づいて、ターゲット命令の実行に必要な即値を拡張することが出来る。

【0039】請求項11の発明は、請求項1～請求項10のいずれかにおいて、前記命令コード解析手段は、プリフィックス命令入力後にターゲット命令を入力した場合、該ターゲット命令のオペレーション内容を拡張して解釈するターゲット命令機能拡張手段を含み、前記命令実行手段は、前記ターゲット命令機能拡張手段で拡張されたオペレーション内容で該ターゲット命令を実行することを特徴とする。

【0040】この様にすると、ターゲット命令をプリフィックス命令と組み合わせることで実行させることにより、ターゲット命令に拡張された機能を実現させることができる。このため、命令数を減らすことができ、さらには命令コードに使用するビット数を減らすことが可能となる。

40 【0041】また、ビット数の少ない命令コードでは、実現が難しい機能を、プリフィックス命令とターゲット命令を組み合わせることにより実現することも可能となる。このため、命令コードの短縮によるメモリの使用効率を改善することができる。

【0042】請求項12の発明は、請求項11において、前記ターゲット命令機能拡張手段は、プリフィックス命令入力後に2オペランド命令である所定のターゲット命令を入力した場合、該ターゲット命令の2つのオペランドとプリフィックス命令に含まれた即値を用いて、該ターゲット命令のオペレーション内容を3オペランド

50

(6)

9

命令に拡張して解釈することを特徴とする。

【0043】この様すると、2オペランド命令のターゲット命令をプリフィックス命令と組み合わせて実行させることにより、該ターゲット命令の操作機能を3オペランド命令に拡張して実行させることができる。従って、2オペランド分記述領域しかない短い命令コードを使用しても3オペランド命令を実現することが出来る。このため、命令数及び命令コードに使用するビット数を減らすことが可能となる。

【0044】また、さらに、プリフィックス命令で即値データを拡張して、3オペランド命令を実現することが出来るため、通常1命令で3オペランド命令を実行する場合に比べて大きな即値データを扱うことができる。

【0045】請求項13の発明は、請求項11において、前記ターゲット命令機能拡張手段は、プリフィックス命令入力後にレジスタ指定値を有する所定のターゲット命令を入力した場合、プリフィックス命令の命令コードに含まれた即値に基づき、ターゲット命令の命令コードに指定されたレジスタに格納されたデータ又はアドレスを修正して実行するようにオペレーション内容を拡張することを特徴とする。

【0046】ここにおいて、データのアドレスを修正して実行するとは、データ又はアドレスに四則演算又は論理演算、シフト演算等を施して実行することをいう。この様すると、ターゲット命令をプリフィックス命令を組み合わせて実行させることにより、ターゲット命令の命令コードで指定されたレジスタに格納されたデータ又はアドレスをプリフィックス命令の命令コードに含まれた即値で修正して実行することが出来るため、命令数又は命令コードに使用するビット数を減らすことが可能となる。

【0047】請求項14の発明は、請求項13において、前記ターゲット命令機能拡張手段は、プリフィックス命令入力後にレジスタ指定値を有する所定のターゲット命令を入力した場合、プリフィックス命令の命令コードに含まれた即値に基づき、ターゲット命令の命令コードに指定されたレジスタに格納されたアドレスのディスプレイースメントを作成し、該ディスプレイースメントを用いて実行するようにオペレーション内容を拡張することを特徴とする。

【0048】このようにすると、プリフィックス命令を用いることにより、実行時のアドレスのディスプレイースメントを作成することが可能となる。

【0049】さらに、プリフィックス命令を用いることにより、オペランドのビット数に限定されずにディスプレイースメントの付加が可能になる。

【0050】請求項15の発明は、請求項1～請求項14のいずれかにおいて、前記命令コード解析手段は、プリフィックス命令の連続入力回数に基づき、プリフィックス命令のオペレーション内容を拡張するプリフィック

10

ス命令機能拡張手段を含むことを特徴とする。

【0051】ここにおいて、複数の種類のプリフィックス命令がある場合は、同種類のプリフィックス命令の連続入力回数に基づき、プリフィックス命令のオペレーション内容を拡張する。ここでの同種類のプリフィックス命令とは、該プリフィックス命令のオペレーション内容が同一である場合をいう。したがって、複数の種類のプリフィックス命令がある場合であって、それらの機能がオペコード等によってことなるような場合は、同種類のプリフィックス命令とは、同じオペコード等を有するものをいう。

【0052】この様すると、プリフィックス命令のオペレーション内容をその入力回数に基づき変更させることが出来るため、少ない種類のプリフィックス命令で多くの機能を実現することが出来る。

【0053】請求項16の発明は、請求項1～請求項15のいずれかにおいて、前記命令コード解析手段に入力する命令コードを記憶する命令コード記憶手段さらに含むことを特徴とする。

【0054】この様にとすると、短くて冗長部分の少ない命令コードが格納された、メモリーの使用効率のよい情報処理装置を実現することができる。従って、同じ大きさであればより多くの命令を格納出来る情報処理装置を提供することが出来、また、同じ機能であれば、少ないメモリ容量で実現することができる。

【0055】請求項17の発明は、請求項1～請求項16記載の情報処理回路を含むことを特徴とする半導体集積回路装置である。

【0056】本発明によれば、小さな回路構成で高機能な半導体集積回路装置を提供することが出来る。

【0057】請求項18の発明は、請求項1～請求項17記載の情報処理回路を含むことを特徴とするマイクロコンピュータである。

【0058】本発明によれば、簡単な構成でメモリの使用効率のよい、マイクロコンピュータを提供することが出来る。

【0059】請求項19の発明は、請求項18において、RISC方式であることを特徴とするマイクロコンピュータである。

【0060】RISC方式のマイクロコンピュータは、固定長の短い命令コードを処理することにより、命令のデコードに要する時間を短縮し、マイクロコンピュータの回路規模を小さくすることを目的としている。従って、本発明によれば、これらの目的を容易に実現出来るRISC方式のマイクロコンピュータを提供することが出来る。

【0061】請求項20の発明は、請求項18又は請求項19記載のマイクロコンピュータを用いて制御することを特徴とする電子機器である。

【0062】本発明によれば、簡単な構成でメモリの使

(7)

11

用効率のよい情報処理回路を内蔵しているため、安価で高機能な電子機器を提供することが出来る。

【0063】

【発明の実施の形態】以下、本発明の実施の形態を図面に基づき説明する。

【0064】（実施の形態1）

1. CPUの機能説明

図1は、本発明の実施の形態のCPUの機能ブロック図である。該CPU30は32ビット幅のデータを扱うが16ビットの命令コードを処理するよう構成されてい

る。

【0065】本CPU30は、命令デコード部40と、命令実行部60とレジスタファイル90を含む。そしてこのCPUは、16ビットの命令データバス130と、命令データアクセスのための命令アドレスバス120と、32ビットのデータバス140と、データアクセスのためのデータアドレスバス310と、コントロール信号のためのコントロールバス390を介して外部と信号のやり取りを行う。

【0066】前記命令デコード部40は、入力した命令コードを解釈し、該命令を実行するために必要な処理を行うのもで、本発明の特徴的な機能であるext命令処理部50を含む。レジスタファイル90は、汎用レジスタR0～R15の16本の汎用レジスタ、プログラムカウンタ（PC）、プロセッサステータスレジスタ（PSR）、スタックポインタ（SP）、算術ローレジスタ（ALR）、算術ハイレジスタ（AHR）等のCPUで使用するレジスタを有している。命令実行部60は、前記命令デコード部40が解析した命令のオペレーション内容に基づき該命令の実行をおこなうもので、データの演算を行うデータ演算部70と、アドレスの演算を行うアドレス演算部80を含む。

【0067】該CPU内部のハードウェアのデータサイズは32ビットであるため、前記命令実行部60のデータ演算部70やアドレス演算部80で行われる算術論理演算や、前記レジスタファイルのレジスタのビット数および、データの転送等はすべて32ビットになる。ところが、前記CPU30が処理する命令コードは16ビットの固定長である。すなわち、命令データバス及び命令アドレスバスを介して接続されている図示しない命令コード記憶部には、固定長16ビットの命令コードが格納されている。従って、1命令中の有効な即値のビット数は、命令コードのうちクラスコードやオペコードに占められるビット数を除いたビット数となる。このため、1命令中の有効な即値のビット数は、例えば6ビットから8ビットという比較的短いビット幅に制限されている。

【0068】そこで、本実施の形態では、ext命令という固定長16ビットの命令を新たに設け、該命令を用いることにより即値のビット幅を拡張させることが出来るよう構成している。また、ext命令を用いることに

12

より固定長16ビットという短いビット幅の命令コードでは記述出来ないオペレーションを可能にするよう構成している。

【0069】ext命令は、それ単独ではCPUにおける実行を何ら行わないが、ターゲット命令の直前におくことにより、そのターゲット命令の機能を拡張する命令であるプリフィックス命令である。なお、該プリフィックス命令による機能拡張の対象となる命令を、ターゲット命令という。

【0070】この様なext命令の処理を行うのがext命令処理部50であり、詳細は後述する。

【0071】2. 半導体集積回路の構成

マイクロコンピュータに内蔵されている半導体集積回路を例にとり、本発明の特徴的な機能を実現するための回路構成の一例と動作内容について説明する。

【0072】図2は、マイクロコンピュータに内蔵されている半導体集積回路の構成の説明に必要な部分を図示したものである。該半導体集積回路100は、CPU30と、ROM110、図示しないRAMを含む。前記CPU30は、前記ROM110、図示しないRAM等と、16ビットの命令データバス130、該命令データアクセスのための命令アドレスバス120、32ビットのデータバス140とを介して、外部と信号のやり取りを行う。

【0073】前記ROM110には、CPU30に実行させる処理を記述した16ビットの命令コードが記憶されており、命令コード記憶手段として機能する。そして、該ROM110に通常の命令の命令コードと同じように、前記プリフィックス命令やターゲットとなる命令の命令コードが記憶される。

【0074】前記CPU30は、命令レジスタ150と、命令デコード回路160と、即値生成回路170と、EXT1レジスタ172、EXT2レジスタ174、レジスタファイル90と、ALU190とを含む。

【0075】命令レジスタ150は、前記ROM110から、命令データバス130を介して入力された命令コードを格納する。命令デコード回路160は、命令レジスタ150に格納された命令コードを解釈し、実行に必要な制御信号を出力する。また、命令コード中の即値を切り出し、必要に応じてEXT1レジスタ172、EXT2レジスタ174に保持したり、即値生成回路170に出力する。

【0076】即値生成回路170は、前記命令デコード回路160から出力された即値及びEXT1レジスタ172、EXT2レジスタ174に保持された即値を用いて、32ビットに拡張された即値を生成する。すなわち、命令デコード回路160と即値生成回路170とEXT1レジスタ172とEXT2レジスタ174は、図1に示された命令デコード部40及びext命令処理部50として機能する。

(8)

13

【0077】ALU190は、前記拡張された即値やレジスタファイル90のレジスタに格納された値や図示しない主記憶のデータに対して、算術演算や論理演算やシフト演算を行う。すなわち、ALU190は図1に示された命令実行部40のデータ演算部70として機能する。

【0078】次に本発明の特徴的な機能である、拡張された即値の生成を行う構成について詳細に説明する。本発明の即値の拡張は後述するよう様々なパターンがあるが、本実施の形態では、プリフィックス命令の即値を用いて、後続のターゲット命令の即値のビット数を増やす拡張を行う場合を例に取り、その詳細な構成を説明する。

【0079】本マイクロコンピュータで処理される命令は、通常の命令と、プリフィックス命令にわけることができる。通常の命令のうち所定の命令については、プリフィックス命令のターゲットとなることができる（プリフィックス命令のターゲットになった命令をターゲット命令という）。

【0080】本マイクロコンピュータでは、16ビット固定長命令コードを採用しているため、後述するように通常の命令では、即値を指定するフィールドとして、例えば6ビットしか確保できない。しかし、ALU190等により実行される際のデータ長は32ビットであるため、前記即値生成回路170で、32ビットに拡張された即値が生成され、ALU190に入力される。この様な場合、通常のビットの拡張としては、以下の2種類がある。すなわち、前記6ビットの上位26ビットをすべてゼロとするゼロ拡張と、前記6ビットの最上位ビットの値を上位26ビットに拡張するサイン拡張である。しかしこの様な拡張では、CPUは32ビットの演算能力があるにもかかわらず、意味のある内容を有する大きなビット数の即値を扱うことが出来ない。従って、本実施の形態では、プリフィックス命令の即値を用いて、後続のターゲット命令の即値のビット数を増やす拡張を行うことができるような構成を採用している。本実施の形態ではこの様な拡張を、プリフィックス命令の命令コードに含まれる即値を保持する手段と、保持された即値に基づき即値のビット数を増やす手段とで実現している。

【0081】まずプリフィックス命令の命令コードに含まれる即値を保持する手段について説明する。該手段として、命令デコード回路160は図示しないステートマシンを有している。該ステートマシンの状態は、命令デコード回路160が処理した命令が、プリフィックス命令であるか、通常の命令であるかによって遷移する。命令デコード回路160は、ステートマシンの状態に基づき、処理したプリフィックス命令の即値をEXT1レジスタ172やEXT2レジスタ174に保持させる。即値生成回路170は、ステートマシンの状態に基づいて前記EXT1レジスタ172やEXT2レジスタ174

14

に保持された即値を用いて、拡張された即値の生成を行う。なお、保持手段は、レジスタのほかラッチ回路でもよい。

【0082】なお、後述するように本実施の形態のマイクロコンピュータで使用するプリフィックス命令はext命令一種類のみであり、ext命令の命令コードは13ビットの即値を含んでいる。

【0083】図3は、命令デコード回路160が有するステートマシンの状態の遷移を表した図である。前記ステートマシンは、S0、S1、S2の3種類の状態を保持することが出来る。S0は、ext命令（プリフィックス命令）でない通常の命令を入力した状態を表しており、S1は、1回目のext命令（プリフィックス命令）を入力した状態をあらわしており、S2は、2回目のext命令（プリフィックス命令）を入力した状態を表している。

【0084】同図に示すように、ステートマシンの状態がS0の時、通常の命令（ext=0）を入力すると状態はS0のままであり（400）、ext命令（ext=1）を入力すると状態はS1に遷移する（410）。また、ステートマシンの状態がS1の時、通常の命令（ext=0）を入力すると状態はS0に遷移し（420）、ext命令（ext=1）を入力すると状態はS2に遷移する（430）。さらに、ステートマシンの状態がS2の時、通常の命令（ext=0）を入力すると状態はS0に遷移し（440）、ext命令（ext=1）を入力すると状態はS2のままである（450）。

【0085】すなわち、前記ステートマシンは、通常の命令が入力されるとS0の状態になり、ext命令を連続して2回以上入力した場合は、再び通常の命令を入力するまでS2の状態を保持するよう構成されている。

【0086】図4は、即値の保持に関して命令デコード回路160が行う動作のアルゴリズムを示したフローチャート図である。

【0087】まず命令デコード回路160は、命令レジスタに入力された命令コードを解釈し、該命令コードがext命令であるか否か及びステートマシンの現在の状態に基づき、ステートマシンの状態設定を行う（ステップ10）。

【0088】ステートマシンの状態がS1であるときは（ステップ20）、1回目ext命令が入力された状態であり、命令デコード回路160は、1回目のext命令の13ビットの即値をext1レジスタ172に保持する（ステップ30）。

【0089】ステートマシンの状態がS2であるときは（ステップ40）、2回以上連続してext命令が入力された状態であり、命令デコード回路160は、入力されたext命令の13ビットの即値をext2レジスタ174に保持する（ステップ50）。このため、ext命令が3回以上連続した場合は、最新のext命令の即

(9)

15

値で `ext2` レジスタが上書きされることになる。従って、`ext` 命令が連続した場合は、最初の `ext` 命令と最後の `ext` 命令の即値が有効となり、その中間の `ext` 命令は無視されることになる。

【0090】そして、状態が `S0` である場合は、通常の命令が入力された状態であるため、該命令の命令コードが即値を含むか否かを判断し（ステップ60）、即値を含む場合は、該命令コードの即値を即値生成回路に出力する（ステップ70）。

【0091】次に、保持された即値に基づき即値のビット数を増やす手段について説明する。該手段として、即値生成回路170は図示しない即値生成用信号生成手段を有しており、該即値生成用信号生成手段で生成された即値生成信号に基づき拡張された即値を生成する。

【0092】図5は、入力される命令コード及び前記ステートマシンの状態と前記即値生成用信号生成回路が生成する即値生成用信号との関係を示したタイミングチャート図である。

【0093】前記即値生成用信号生成回路が生成する信号は `ext` 信号530と、`ext_low` 信号550と `ext_high` 信号560である。図5の510は入力される命令コードを表しており、540は前記ステートマシンの状態を表している。またクロック信号520は、図2において図示しないクロック信号発生器で生成されるか又はクロック入力端子から入力される信号である。該クロック信号520は、CPUにおける各種動作の同期をとるために用いられる。例えば、該クロック信号520の立ち上がりに同期して命令アドレスバス120に命令アドレスが出力される。また、該クロック信号520の1周期（1マシンサイクル）毎に前記命令アドレスに基づいてROM110から命令コードが読み出され、命令レジスタ150に保持される。そして、読み出された命令コードに応じたオペレーションは、1マシンサイクル内で完了する。

【0094】`ext` 信号530は、`ext` 命令が入力されたときに '1' となる信号であって、前記ステートマシンの状態遷移を生じさせる入力信号である。すなわち、前記即値生成用信号生成回路は、入力された命令が `ext` 命令である場合 `ext` 信号530を '1' にし、入力された命令が通常の命令である場合 `ext` 信号530を '0' にするよう動作する。

【0095】`ext_low` 信号550は、ステートマシンの状態に応じて出力される信号であり、状態が 'S1' の時 '1' となる1クロック分遅延した遅延信号である。すなわち、前記即値生成用信号生成回路は、状態が 'S1' である時、次のクロック信号の立ち上がりに同期をとって、`ext_low` 信号550を '1' にし、その次のクロック信号の立ち上がりに同期をとって、`ext_low` 信号550を '0' とする。

【0096】`ext_high` 信号560は、ステート

16

マシンの状態に応じて出力される信号であり、状態が 'S2' の時 '1' となる1クロック分遅延した遅延信号である。すなわち、前記即値生成用信号生成回路は、状態が 'S2' である時、次のクロック信号520の立ち上がりに同期をとって、`ext_high` 信号560を '1' にし、その次のクロック信号520の立ち上がりに同期をとって、`ext_high` 信号560を '0' とする。

【0097】図6は、命令コードに即値を含んでいる場合に即値生成回路が、ステートマシンの状態及び即値生成用信号に基づき即値を拡張するアルゴリズムを示したフローチャート図である。

【0098】ステートマシンの状態が `S0` で無い場合は（ステップ110）、CPUがプリフィックス命令である `ext` 命令を処理している状態であり、通常の命令を処理している状態ではない。この場合には、即値生成回路170は、即値の生成を行わない（ステップ120）。プリフィックス命令自体は、命令実行部60におけるALUでの演算等の実行は行われなため、プリフィックス命令の処理時（ステートマシンの状態が `S1` 又は `S2` の場合）に即値を生成する必要はないからである。

【0099】ステートマシンの状態が `S0` で `ext_low` 信号550 = '1' の場合は（ステップ110）、一回の `ext` 命令の後にターゲット命令である通常の命令を処理している状態である。この場合には、EXT1レジスタ172に保持されている先の `ext` 命令の命令コードに含まれていた13ビットの即値と、即値生成用信号線176を介して入力された該ターゲット命令の命令コードに含まれていた6ビットの即値をつなげて出来る19ビットの即値を、32ビットにゼロ拡張又はサイン拡張する（ステップ140）。

【0100】ステートマシンの状態が `S0` で `ext_high` 信号560 = '1' の場合は（ステップ150）、複数回連続して入力した `ext` 命令の後にターゲット命令である通常の命令を処理している状態である。この場合には、EXT1レジスタ172に保持されている一番先の `ext` 命令の命令コードに含まれていた13ビットの即値と、EXT2レジスタ174に保持されている一番最後の `ext` 命令の命令コードに含まれていた13ビットの即値と、即値生成用信号線176を介して入力された該ターゲット命令の命令コードに含まれていた6ビットの即値とをつなげて32ビットの即値を生成する（ステップ160）。

【0101】ステートマシンの状態が `S0` で `ext_low` 信号550 = '0' かつ `ext_high` 信号560 = '0' の場合は、ターゲット命令でない通常の命令（直前の命令が `ext` 命令でない通常の命令）を処理している状態である。この場合には、即値生成用信号線176を介して入力された該通常の命令の命令コードに含

(10)

17

まれていた6ビットの即値を32ビットにゼロ拡張又はサイン拡張する(ステップ170)。

【0102】この様に、命令デコード回路160および即値生成回路170はext命令の即値を用いて、後続のターゲット命令の即値のビット数を増やす即値拡張手段として機能する。

【0103】なお、前記処理は、命令コードに即値を含んでいる場合である。しかし、命令コード中に即値を含んでいない場合でも、ステップ140の処理とステップ160の処理で命令コード中の即値を用いないようにし、ステップ170の処理を省くことにより前記処理と同様に適用出来る。

【0104】ext命令を連続して複数回実行することにより、2回目以降のext命令のオペレーション内容を拡張することが出来る。すなわち、前述したような構成をとることにより、命令デコード回路160と即値生成回路170とEXT1レジスタ172とEXT2レジスタ174は、プリフィックス命令の連続入力回数に基づき、該プリフィックス命令のオペレーション内容を拡張して解釈するプリフィックス命令機能拡張手段として機能する。

【0105】3. ext命令(プリフィックス命令)を用いたターゲット命令の即値拡張の具体例
本発明のext命令による即値拡張の代表的な実施の形態を、命令コード中に即値と汎用レジスタを指定し、該即値と汎用レジスタに格納されている値の演算を行うタイプの命令(便宜上、以下タイプ1命令という)の場合を例に取り説明する。

【0106】まず、タイプ1命令及びext命令の命令コードの構成について説明する。図7(A)は、該タイプ1命令の命令コード210のビットフィールドを示した図であり、図7(B)は、ext命令の命令コード220のビットフィールドを示している。ビットフィールドの上の数字はビットの位置を示しており、同図(A)(B)に示すように、命令コードはビット15からビット0まで16ビット幅のフィールドを有している。

【0107】同図(A)に示すように、タイプ1命令の命令コード210は、ビット15からビット13に3ビットのクラス指定領域212と、ビット12からビット10に3ビットのオペコードを指定する領域214と、ビット9からビット4に6ビットの即値を指定する領域216と、ビット3からビット0に4ビットのレジスタ指定領域218を有している。

【0108】クラス指定領域212とは、命令のグループを定義するフィールドである。本マイクロコンピュータで使用する命令は、所定のルールに従って、グループ分けされているため、どのグループに属する命令であるのかがクラス指定領域に指定されている。

【0109】オペコードを指定する領域214には、命令の操作機能を規定するオペコードが格納されている。

18

なお、クラス毎にそのクラスに属するオペコードが決まっているので、該オペコードを指定する領域214には、前記クラス指定領域212で指定されたクラスに属するオペコードが格納されている。

【0110】また、前記即値を指定する領域216には、6ビットの即値(imm6)の値が格納されており、レジスタ指定領域218には、いずれかの汎用レジスタ(rd)を示すコードが格納されている。

【0111】タイプ1命令は、前記即値(imm6)と汎用レジスタ(rd)に対して前記オペコードで示される演算を行い、結果を汎用レジスタ(rd)に対して書き込む動作を行う命令である。

【0112】また、同図(B)に示すように、ext命令の命令コード220は、ビット15からビット13に3ビットのクラス指定領域222と、ビット12からビット0に13ビットの即値を指定する領域224とを有している。前記即値を指定する領域224には、13ビットの即値(imm13)の値が格納されている。

【0113】該ext命令はオペコードを有していないが、本実施の形態のマイクロコンピュータが処理するプリフィックス命令は、ext命令1種類のみであり、所定のクラスを該プリフィックス命令に割り当てているため、クラス指定領域222に指定されたクラスによって、ext命令であると判別することが出来る。

【0114】ext命令はプリフィックス命令であるため、単独ではCPUにおけるALUでの演算等の実行を何ら行わないが、後続のターゲット命令が実行される際に、そのターゲット命令の実行に使用する即値を拡張する機能を有する。例えばタイプ1命令のように命令コードに即値を含んでいる命令がext命令のターゲット命令となった場合、タイプ1命令実行時に、タイプ1命令の命令コードに含まれる即値をext命令の6ビットの即値(imm6)を用いて拡張する機能を有する。

【0115】次に前記タイプ1命令が実行される際の即値の拡張について詳しく説明する。

【0116】まず、タイプ1命令が単独で実行された場合の実行内容について説明する。図2において、まず、タイプ1命令はROM110から命令データバス120を介して命令レジスタ150に入力される。そして、命令でコード回路160で解釈され、内容に応じた演算がおこなわれる。タイプ1命令においては、命令コードで指定された汎用レジスタ(rd)に格納されたデータがレジスタファイル90よりデータバス182を介して、ALU190に入力される。また、命令コードで指定された即値(imm6)は、命令デコード回路により切り出され、即値生成用信号線176を介して即値生成回路170に入力される。そして、即値生成回路170で、前記即値(imm6)は32ビットにゼロ拡張あるいはサイン拡張され、前記拡張された即値はALU190に入力される。そしてALU190はタイプ1命令のオペ

(11)

19

コードで示される演算を行い、演算結果をデータバス192を介して、レジスタファイル90の汎用レジスタ(r d)に格納する。

【0117】前記タイプ1命令は単独で実行することも出来るし、直前の1又は複数のe x t命令と組み合わせて実行することも可能である。e x t命令と組み合わせて実行された場合の実行内容についても、基本的には前記単独実行の場合と同様であるが、命令デコード回路により切り出された即値(i m m 6)から拡張された即値が生成される過程が異なってくる。すなわち、e x t命令と組み合わせて実行された場合は、E X T 1レジスタ172やE X T 2レジスタ174に保持されている先のe x t命令の即値を用いてi m m 6の拡張が行われる。この過程および、この過程に先だってe x t命令の即値がE X T 1レジスタ172やE X T 2レジスタ174に保持される過程は、図3～図6で説明した通りである。従って、タイプ1命令が単独で実行されたか、直前の1又は複数のe x t命令と組み合わせて実行されたかにより、即値生成回路170で、生成される拡張された即値が異なってくる。

【0118】図8(A)～(C)は、前記各場合のオペレーションを表した式と該オペレーションの実行に使用される拡張された即値のフィールド図である。

【0119】図8(A)は、タイプ1命令が単独で実行された場合のオペレーションを表した式と、実行に使用される拡張された即値230のビットフィールド図である。同図に示すように、タイプ1命令の6ビットの即値(i m m 6)がゼロ拡張又はサイン拡張のいずれかの方法で拡張されて32ビットの即値230となる。ゼロ拡張された場合は、ビット6からビット31の領域232はすべてゼロとなり、サイン拡張された場合は、ビット6からビット31の領域232は、すべてi m m 6の最上位ビットすなわちビット5と同じビットとなる。

【0120】図8(B)は、直前の1のe x t命令と組み合わせて実行された場合のオペレーションを表した式と、該オペレーションの実行に使用される拡張された即値240のビットフィールド図である。同図に示すように、タイプ1命令の6ビットの即値(i m m 6)がビット5からビット0のフィールド246にセットされ、直前の1のe x t命令の13ビットの即値(i m m 13)がビット18からビット6にセットされ、19ビットの即値(i m m 19)が生成される。そして、前記19ビットの即値(i m m 19)はゼロ拡張又はサイン拡張のいずれかの拡張方法で32ビットの即値240となる。ゼロ拡張された場合は、19ビットから31ビットの領域242はすべてゼロとなり、サイン拡張された場合は、19ビットから31ビットの領域242は、すべてi m m 19の最上位ビットすなわち18ビット目と同じビットとなる。

【0121】図8(C)は、2回のe x t命令と組み合

20

わせて実行された場合のオペレーションを表した式と、該オペレーションの実行に使用される拡張された即値250のビットフィールド図である。同図に示すように、タイプ1命令の6ビットの即値(i m m 6)がビット5からビット0のフィールド256にセットされ、1回目のe x t命令の13ビットの即値(i m m 13)がビット31からビット19にセットされ、2回目のe x t命令の13ビットの即値(i m m 13)がビット18からビット6にセットされ、32ビットの即値(i m m 32)250が生成される。

【0122】なお、2回以上のe x t命令と組み合わせて実行場合は、最初と、最後のe x t命令が有効となるよう構成されている。すなわち、最初のe x t命令が前記1回目のe x t命令と同様に機能し、最後のe x t命令が前記2回目のe x t命令と同様に機能する。

【0123】4. プリフィックス命令を用いたターゲット命令のオペレーション内容拡張の具体例
本発明のプリフィックス命令によるターゲット命令のオペレーション内容拡張の代表的な実施の形態を、2汎用レジスタを用いて演算を行うタイプの命令(便宜上、以下タイプ2命令という)の場合を例に取り説明する。

【0124】まず、タイプ2命令の命令コードの構成について説明する。図9は、該タイプ2命令の命令コード260のビットフィールドを示した図である。

【0125】同図に示すように、タイプ2命令の命令コード260は、ビット15からビット13に3ビットのクラス指定領域262と、ビット12からビット8に5ビットのオペコードを指定する領域264と、ビット7からビット4に4ビットのレジスタ指定領域266と、ビット3からビット0に4ビットのレジスタ指定領域268とを有している。

【0126】また、前記レジスタ指定領域266、268には、それぞれいずれかの汎用レジスタ(r s)(r d)を示すコードが格納されている。r sとはレジスタソース、r dとはレジスタディスティネーションを意味するものある。タイプ2命令は、汎用レジスタ(r s)に格納されているデータと汎用レジスタ(r d)に格納されているデータにたいしてオペコードで示される演算を行い、結果を汎用レジスタ(r d)に書き込む動作を行う2オペランド命令である。

【0127】次に、タイプ2命令が単独で実行された場合のCPUの動作について説明する。図2において、まず、タイプ2命令はROM1:10から命令データバス130を介して命令レジスタ150に入力される。そして、命令デコード回路160で解読され、内容に応じた演算がおこなわれる。タイプ2命令においては、命令コードで指定された汎用レジスタ(r d)に格納されたデータがレジスタファイル90よりデータバス182を介して、ALU190に入力される。また、命令コードで指定された汎用レジスタ(r s)に格納されたデータが

(12)

21

レジスタファイル90よりデータバス184を介して、ALU190に入力される。そしてALU190はタイプ2命令のオペコードで示される演算を行い、演算結果をデータバス192を介して、レジスタファイル90の汎用レジスタ(r d)に格納する。

【0128】前記タイプ2命令は単独で実行することも出来るし、直前の1又は複数のe x t命令と組み合わせて実行することも可能である。

【0129】図10(A)～(C)は、タイプ2命令を単独で実行する場合及び直前の1又は2回のe x t命令と組み合わせて実行する場合のオペレーションを表した式と、実行に使用する即値のビットフィールドを示した図である。

【0130】図10(A)は、タイプ2命令が単独で実行された場合のオペレーションを表した式である。単独で実行される場合は、同式に示すように、タイプ2命令の本来のオペレーションが実行されるため、即値は使用されない。なお、タイプ2命令が単独で実行された場合の実行内容については前述した通りである。

【0131】図10(B)は、タイプ2命令が直前の1のe x t命令と組み合わせて実行された場合のオペレーションを表した式と、該オペレーションの実行に使用される拡張された即値270のビットフィールド図である。同式に示すように、タイプ2命令が直前の1のe x t命令と組み合わせて実行された場合は、e x t命令に基づき生成される即値(i m m 1 3)と、汎用レジスタ(r s)の内容との演算結果を、汎用レジスタ(r d)に書き込むようにオペレーション内容が拡張する。即値270は、e x t命令に含まれる即値(i m m 1 3)が32ビットにゼロ拡張して生成される。

【0132】タイプ2命令が1のe x t命令と組み合わせて実行された場合の実行内容について説明する。図2において、まず、タイプ2命令はROM110から命令データバス130を介して命令レジスタ150に入力される。そして、命令デコード回路160で解読されるが、このとき該命令がターゲット命令になっている場合(前記ステートマシンの状態がS0以外の場合)は、以下のオペレーションを実行する。すなわち、命令コードで指定された汎用レジスタ(r s)に格納されたデータがレジスタファイル90よりデータバス182を介して、ALU190に入力される。また、EXT1レジスタ172に保持されていた直前のe x t命令の即値(i m m 1 3)は、即値生成回路170によりゼロ拡張されて前記即値270が生成され、ALU190に入力される。そして、ALU190はタイプ2命令のオペコードで示される演算を行い、演算結果をデータバス192を介して、レジスタファイル90の汎用レジスタ(r d)に格納する。

【0133】なお、EXT1レジスタ172やEXT2レジスタ174に保持されている先のe x t命令の即値

22

を用いて、タイプ2命令の実行に使用する即値を生成す過程および、この過程に先だつてe x t命令の即値がEXT1レジスタ172やEXT2レジスタ174に保持される過程は、図3～図6で説明した通りである。

【0134】従って、タイプ2命令をe x t命令と組み合わせて実行した場合、オペランドに即値と2つの汎用レジスタとを有するような3オペランド命令を実行したのと同様の効果を有する。

【0135】図10(C)は、タイプ2命令が2回のe x t命令と組み合わせて実行された場合のオペレーションを表した式と、該オペレーションの実行に使用される拡張された即値280のビットフィールド図である。同式に示すように、タイプ2命令が2回のe x t命令と組み合わせて実行された場合は、2回のe x t命令に基づき生成される即値(i m m 2 6)と、汎用レジスタ(r s)の内容との演算結果を、汎用レジスタ(r d)に書き込むようにオペレーション内容が拡張する。即値280は、1回目のe x t命令に含まれている即値i m m 1 3をビット25からビット13にセットし、2回目のe x t命令に含まれている即値をビット12からビット0にセットしたi m m 2 6を32ビットにゼロ拡張又はサイン拡張して生成される。

【0136】なお、2回以上のe x t命令と組み合わせて実行された場合は、最初と、最後のe x t命令が有効となるよう構成されている。すなわち、最初のe x t命令が前記1回目のe x t命令と同様に機能し、最後のe x t命令が前記2回目のe x t命令と同様に機能する。

【0137】タイプ2命令が2回または複数回のe x t命令と組み合わせて実行された場合の実行内容は、基本的には、1回のe x t命令と組み合わせて実行された場合と同様であるが、即値生成回路170が前記即値280を生成する過程がことになって来る。

【0138】すなわち、即値生成回路170は、EXT1レジスタ172に保持されていた1回目のe x t命令の即値(i m m 1 3)と、EXT2レジスタ174に保持されていた最後のe x t命令の即値(i m m 1 3)を用いて前記即値280を生成する。

【0139】なお、EXT1レジスタ172やEXT2レジスタ174に保持されている先のe x t命令の即値を用いて、タイプ2命令の実行に使用する即値を生成す過程および、この過程に先だつてe x t命令の即値がEXT1レジスタ172やEXT2レジスタ174に保持される過程は、図3～図6で説明した通りである。

【0140】この様に、e x t命令を用いることにより、ターゲット命令のオペレーション内容を拡張することができるため、命令デコード回路及び即値生成回路は、ターゲット命令機能拡張手段として機能する。

【0141】また、タイプ2命令を2回又は複数回のe x t命令と組み合わせて実行した場合、2つのe x t命令の13ビットの即値(i m m 1 3)をつなげたものをゼ

(13)

23

ロ拡張又はサイン拡張した即値を使用できるため、1つのext命令と組み合わせた場合よりも大きな有効ビット数を有する即値を処理することが出来る。

【0142】この様に、タイプ2命令を2回又は複数回のext命令と組み合わせて実行した場合、1回のext命令と組み合わせて実行した場合にくらべて機能が拡張するため、命令デコード回路及び即値生成回路は、プリフィックス命令機能拡張手段として機能する。

【0143】5. プリフィックス命令を用いたターゲット命令のアドレスの即値拡張の具体例

次に本発明のプリフィックス命令によるターゲット命令実行時のアドレスの拡張について説明する。図11は、前記マイクロコンピュータに内蔵されている半導体集積回路の構成の説明に必要な部分を図示したものである。基本的には、図2に示す構成と同様であり、図2と同様の機能を有する部分については、同符号を付している。なお、簡単のため、図2のEXT1レジスタ172等は省略して、命令デコード回路160と即値生成回路170のみ図示している。また、図2の構成に加えて、RAM320、プログラムカウンタPC330、アドレス演算器340、Xバス350を図示している。なお、説明の都合上、プログラムカウンタPC330をレジスタファイル90とは別に図示している。また、アドレス演算器340は加減乗除等の演算を行うが、本具体例では、便宜上加算演算をする機能のみを取り上げ説明する。

【0144】前記RAM320は、実行に使用するデータ等を記憶している。アドレス演算器340は、前記RAM320に記憶されているデータのアドレスを演算するものであり、即値生成回路170で生成された即値及びプログラムカウンタPC330の値もしくはレジスタファイル90に格納された値を入力するよう構成されている。プログラムカウンタPC330には、現在実行中の命令のアドレスが格納されている。そして、前記アドレス演算器340で演算されたアドレスによりデータアドレスバス310を介して、RAM320のアドレスを指定し、又は、命令アドレスバス120を介して、ROM110の命令のアドレスを指定する。そして、前記各アドレスバス310、120で指定されたアドレスのデータ又は命令コードは、データバス140又は命令データバス130を介してCPU30に入力されるよう構成されている。

【0145】次に、オペランドにアドレスの即値を有し、プログラムカウンタPC330に格納されている値に該即値を加算した相対アドレスを実行に使用するタイプの命令（便宜上、以下タイプ3命令という）におけるプリフィックス命令によるアドレスの即値の拡張を説明する。

【0146】図12は、タイプ3命令の一例であるPC相対サブルーチンコール命令（相対アドレスでサブルーチンをコールする命令）の命令コード290のビットフ

24

ィールドを示した図である。

【0147】同図に示すように、タイプ3命令の命令コード290は、ビット15からビット13に3ビットのクラス指定領域292と、ビット12からビット9に4ビットのオペコードを指定する領域294と、ビット8の所定のオペコードに付帯したビットを指定するdビット領域296と、ビット7からビット0に8ビットの即値を指定する領域298を有している。

【0148】PC相対サブルーチンコール命令は、前記即値を指定する領域298に格納された8ビットの即値（imm8）を1ビット左にシフトすることで9ビットの値（sin9）とし、該sin9の値をサイン拡張して得られる32ビットの即値とPC（プログラムカウンタ）の値とを加算した値をアドレスとして該アドレスに分岐する命令である。

【0149】図11をもちいて、該PC相対サブルーチンコール命令が単独で実行される場合のCPUの動作について説明する。即値生成回路170によって、該命令コードに含まれる8ビットの即値は、1ビット左にシフトされ、サイン拡張され32ビットの即値（sign9）となる。そして、該32ビットの即値（sign9）はアドレス演算器340に入力される。また、プログラムカウンタPC330に格納されているアドレスの値もアドレス演算器340に入力される。アドレス演算器340は、前記即値（sign9）とプログラムカウンタPC330に格納されているアドレスの値を加算して分岐アドレスを生成する。そして、分岐アドレスに格納されたサブルーチンが実行されることになる。

【0150】PC相対サブルーチンコール命令のようなタイプ3命令は、単独で実行することも出来るし、直前の1又は複数のext命令と組み合わせて実行することも可能である。ext命令と組み合わせて実行された場合の実行内容についても、基本的には前記単独実行の場合と同様であるが、命令デコード回路160の即値生成回路170で、命令コードより切り出された即値（imm8）から拡張された即値が生成される過程が異なってくる。すなわち、ext命令と組み合わせて実行された場合は、EXT1レジスタ172やEXT2レジスタ174に保持されている先のext命令の即値を用いてimm8の拡張が行われる。この過程および、この過程に先だってext命令の即値がEXT1レジスタ172やEXT2レジスタ174に保持される過程は、図3～図6で説明した通りである。従って、タイプ3命令が単独で実行されたか、直前の1又は複数のext命令と組み合わせて実行されたかにより、分岐先のアドレスを定めることが出来る。

【0151】図13(A)～(C)は、前記各場合のタイプ3命令のオペレーションを表した式と該オペレーションの実行に使用される拡張された即値のフィールド図である。

(14)

25

【0152】図13(A)は、タイプ3命令が単独で実行された場合のオペレーションを表した式と、実行に使用される拡張された即値400のビットフィールド図である。CPU30は、16ビット固定長の命令コードを用いているため、命令アドレスの値は常に偶数バイトアドレスとなり、その結果、命令アドレスの最下位ビットは常に"0"となる。そのため、拡張の際には、同図に示すように、タイプ3命令の8ビットの即値(imm8)は1ビット左にシフトした9ビットの即値となる。そして該9ビットの即値はサイン拡張されて32ビットの即値400(sign9)となる。

【0153】図13(B)は、タイプ3命令が直前の1のext命令と組み合わせて実行された場合のオペレーションを表した式と、該オペレーションの実行に使用される拡張された即値410のビットフィールド図である。同図に示すように、タイプ3命令の8ビットの即値(imm8)とext命令の13ビットの即値(imm13)を結合した21ビットの即値は、1ビット左にシフトされて、22ビットの即値となる。該22ビットの即値は、サイン拡張されて32ビットの即値410(sign22)となる。

【0154】図13(C)は、タイプ3命令が2回のext命令と組み合わせて実行された場合のオペレーションを表した式と、該オペレーションの実行に使用される拡張された即値420のビットフィールド図である。同図に示すように、タイプ3命令の8ビットの即値(imm8)と1回目のext命令の10ビットの即値(imm10)と、2回目のext命令の13ビットの即値(imm13)とを結合した31ビットの即値は、1ビット左にシフトされて、32ビットの即値(sign32)となる。

【0155】なお、2回以上のext命令と組み合わせて実行場合は、最初と、最後のext命令が有効となるよう構成されている。すなわち、最初のext命令が前記1回目のext命令と同様に機能し、最後のext命令が前記2回目のext命令と同様に機能する。

【0156】6. プリフィックス命令を用いてターゲット命令の命令中にディスプレースメントを付加する具体例

次に、オペランドには即値を有さず、オペランドに指定したレジスタに格納された値をアドレスとして用いて実行を行うタイプの命令(便宜上、以下タイプ4命令という)において、プリフィックス命令を用いてアドレスのディスプレースメントを付加する例を説明する。

【0157】命令コードの構成は図9に示すタイプ2命令と、2個のレジスタの指定領域をオペランドに有する点で共通する。オペコードの内容が異なってくるが、本実施の形態の説明において特に影響は無いので説明を省略する。タイプ4命令の一例であるロード命令を例にとり、説明する。ロード命令は、rsで指定されるアドレ

26

スに記憶されているデータをrdに示すレジスタにロードするための命令である。

【0158】まず、該ロード命令が単独で実行される場合のCPUの動作について説明する。CPUは、命令コードで指定された汎用レジスタ(rs)に格納された値をアドレスとするデータをRAM320から読み出し、命令コードで指定された汎用レジスタ(rd)に格納する。

【0159】ロード命令のようなタイプ4命令は、単独で実行することも出来るし、直前の1又は複数のext命令と組み合わせて実行することも可能である。ext命令と組み合わせて実行された場合は、ext命令の命令コードに含まれる即値によって、ディスプレースメントが付加されて、タイプ4命令が実行される。

【0160】図14(A)(B)は、タイプ4命令を直前の1又は2回のext命令と組み合わせて実行する場合に生成されるディスプレースメントを示した図である。

【0161】図14(A)は、タイプ4命令が1回のext命令とくみあわされて実行された場合のディスプレースメントを表した図である。同図に示すように、直前の1のext命令の即値(imm13)をゼロ拡張して32ビット即値(imm13)430が生成される。

【0162】タイプ4命令が1のext命令と組み合わせて実行された場合の実行内容について図11を用いて説明する。まず、タイプ4命令はROM110から命令データバス130を介して命令レジスタ150に入力される。そして、命令デコード回路160で解読されるが、このとき該命令がターゲット命令になっている場合(前記ステートマシンの状態がS0以外の場合)は、命令コードで指定された汎用レジスタ(rs)に格納されたアドレスをレジスタファイル90から取り出してXバス350を介して、アドレス演算器340に入力する。

【0163】また、EXT1レジスタ172に保持されていた直前のext命令の即値(imm13)は、即値生成回路170によりゼロ拡張されて32ビットの即値430が生成され、アドレス演算器340に入力される。アドレス演算器340はこれらの2つの入力を加算してアドレスを生成する。生成されたアドレスによりデータアドレスバス310を介して、RAM320のアドレスを指定し、該アドレスに格納されたデータをデータバス140を介してレジスタファイル90に入力し、命令コードで指定された汎用レジスタ(rd)に格納する。

【0164】図14(B)は、タイプ4命令が2回のext命令とくみあわされて実行された場合のディスプレースメントを表している。同図に示すように、1回目のext命令の即値(imm13)をビット25からビット13の13ビットのフィールド444にセットし、2回目のext命令の即値(imm13)をビット12から

(15)

27

ビット0の13ビットのフィールド446にセットし、ゼロ拡張して32ビットの即値440 (imm26) が生成される。

【0165】タイプ4命令が2回のext命令と組み合わせて実行された場合の実行内容について図11を用いて説明する。この場合も1回のext命令と組み合わせた場合と基本的には同様であるが、ディスプレイスメントとして生成される即値が異なってくる。2回のext命令と組み合わせた場合は、EXT1レジスタ172及びEXT2レジスタ174に保持されていた2回のext命令の即値を用いて、図14(B)に示すような即値440が生成され、該即値440がディスプレイスメントとしてアドレス演算器340に入力される。

【0166】なお、EXT1レジスタ172やEXT2レジスタ174に保持されている先のext命令の即値を用いて、タイプ4命令の実行に使用する即値を生成す過程および、この過程に先だってext命令の即値がEXT1レジスタ172やEXT2レジスタ174に保持される過程は、図3～図6で説明した通りである。

【0167】従って、タイプ4命令をext命令と組み合わせて実行した場合、オペランドにディスプレイスメントを付加して実行するようにオペレーション内容が拡張する。

【0168】(実施の形態2) 図15は、本実施の形態のマイクロコンピュータのハードウェアブロック図である。

【0169】該マイクロコンピュータ10は、32ビットマイクロコントローラであり、CPU30とROM110とRAM320、高周波発信回路710、低周波発信回路720、リセット回路730、ブリスケーラ740、16ビットプログラマブルタイマ750、8ビットプログラマブルタイマ760、クロックタイマ770、インテリジェントDMA780、高速DMA790、割り込みコントローラ600、シリアルインターフェース610、バスコントロールユニット(BCU)620、A/D変換器630、D/A変換器640、入力ポート660、出力ポート660、I/Oポート670、及びそれらを接続する各種バス680等、各種ピン690等を含む。

【0170】前記CPU30は、16ビット固定長命令コードを処理し、実行時のデータサイズは32ビットのCPUである。該CPU30は、前述した、実施の形態1の構成を有しており、命令コード解析手段、即値拡張手段、ターゲット命令機能拡張手段、プリフィックス命令機能拡張手段、命令実行手段として機能する。

【0171】従って、16ビットの命令コードを使用しても、大きな即値の取り扱いや、短い命令コードで記述出来ない3オペランド命令の実行を行うことが出来るため、メモリの使用効率のよいマイクロコンピュータを提供することが出来る。また、該マイクロコンピュータを

28

半導体集積回路として構成する場合、記憶させる命令数が同一であれば、32ビット幅の固定長命令コードを使用する場合に比べ、容量は半分でよい。従って、チップのサイズを小さくすることができ歩留まりのよい半導体集積回路装置を製造することが出来る。

【0172】本発明のマイクロコンピュータは例えばプリンター等のパソコン周辺機器や、携帯機器等の各種の電子機器に適用可能である。この様にすると、簡単な構成でメモリの使用効率のよい情報処理回路を内蔵することができるため、安価で高機能な電子機器を提供することが出来る。

【0173】なお本発明は、上記実施例で説明したものに限らず、種々の変形実施が可能である。

【0174】例えば上記実施例では、メモリの使用効率を高めるうえで特に有効な例、すなわち32ビットのデータが処理可能なCPU又はマイクロコンピュータにおいて、16ビットの固定長命令コードを使用する場合を例に取り説明したがこれに限られない。CPU又はマイクロコンピュータで処理可能なビット数及び命令コードのビット数によらず本発明を適用することが出来る。また、命令コードが固定長であるか否かによらず本発明を適用することが出来る。

【0175】そして、本発明を適用することにより、命令コードの即値の拡張及び、オペレーション内容の変更を簡単な構成で容易に行えるCPU又はマイクロコンピュータを提供することが出来る。

【0176】また、上記本実施の形態では、ext命令を2回以上連続実行した場合は1回目と最後のext命令のみ有効に機能する場合を説明したが、3回以上のext命令を有効にするよう構成してもよい。また、有効にするext命令の選択は1回目と最後に限られず所定のルールに従ってさだめればよい。

【0177】また、複数のext命令を組み合わせることによりext命令の機能を変化させる例も上記実施例に限られない。

【0178】また、上記本実施の形態では、プリフィックス命令がext命令1種類のみである場合を説明したが、複数の異なる種類のプリフィックス命令を設定してもよい。

【0179】

【図面の簡単な説明】

【図1】本発明の実施の形態のCPUの機能ブロック図である。

【図2】マイクロコンピュータに内蔵されている半導体集積回路の構成の説明に必要な部分を図示したものである。

【図3】命令デコード回路が有するステートマシンの状態の遷移を表した図である。

【図4】即値の保持に関して命令デコード回路が行う動作のアルゴリズムを示したフローチャート図である。

(16)

29

【図5】入力される命令コード及び前記ステートマシンの状態と前記即値生成用信号生成回路が生成する即値生成用信号との関係を示したタイミングチャート図である。

【図6】即値生成回路が、ステートマシンの状態及び即値生成用信号に基づき即値を拡張するアルゴリズムを示したフローチャート図である。

【図7】同図 (A) (B) は、タイプ1命令及びext命令の命令コードのビットフィールドを示した図である。

【図8】同図 (A) ~ (C) は、オペレーションを表した式及びオペレーションの実行に使用される拡張された即値のフィールド図である。

【図9】タイプ2命令の命令コードのビットフィールドを示した図である。

【図10】同図 (A) ~ (C) は、タイプ2命令のオペレーションを表した式と、実行に使用する即値のビットフィールドを示した図である。

【図11】マイクロコンピュータに内蔵されている半導体集積回路の構成の説明に必要な部分を図示したものである。

【図12】タイプ3命令の命令コードのビットフィールドを示した図である。

【図13】同図 (A) ~ (C) は、タイプ3命令のオペレーションを表した式と該オペレーションの実行に使用される拡張された即値のフィールド図である。

【図14】同図 (A) (B) は、タイプ4命令を実行する場合に生成されるディスプレイメントを示した図で

30

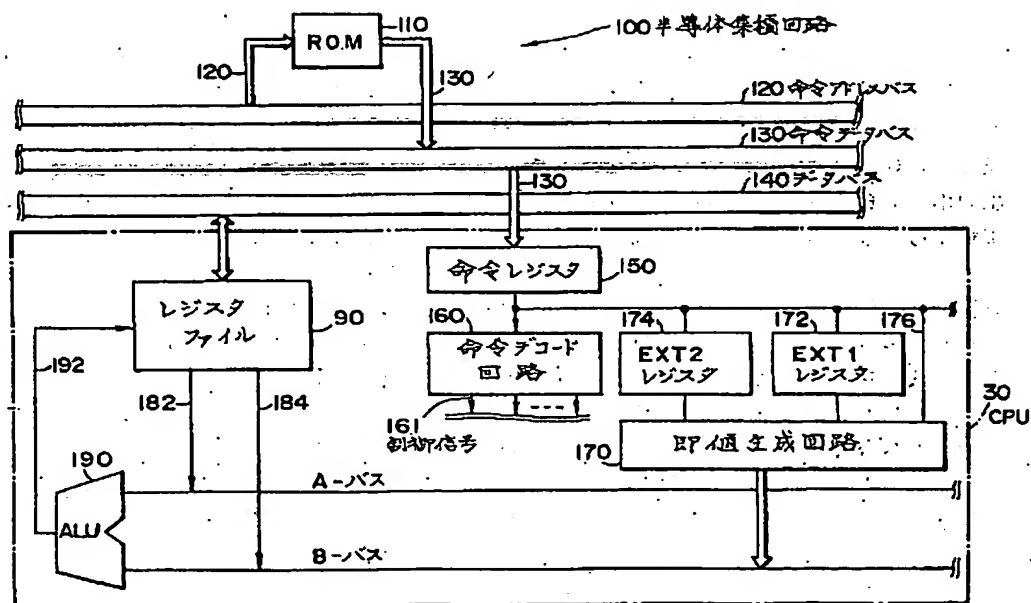
ある。

【図15】実施の形態2のマイクロコンピュータのハードウェアブロック図である。

【符号の説明】

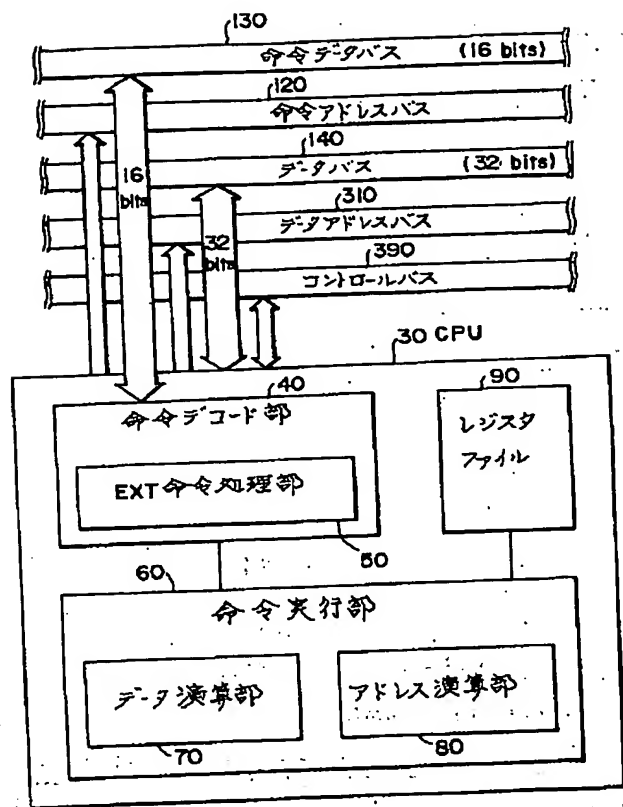
- 10 マイクロコンピュータ
- 30 CPU
- 40 命令デコード部
- 50 ext命令処理部
- 60 命令実行部
- 70 データ演算部
- 80 アドレス演算部
- 90 レジスタファイル
- 110 ROM
- 120 命令アドレスバス
- 130 命令データバス
- 140 データバス
- 150 命令レジスタ
- 160 命令デコード回路
- 161 制御信号
- 170 即値生成回路
- 172 EXT1レジスタ
- 174 EXT2レジスタ
- 190 ALU
- 310 データアドレスバス
- 320 RAM
- 330 プログラムカウンタPC
- 340 アドレス演算器

【図2】

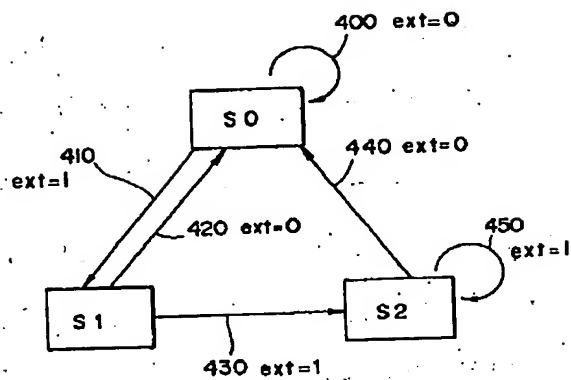


(17)

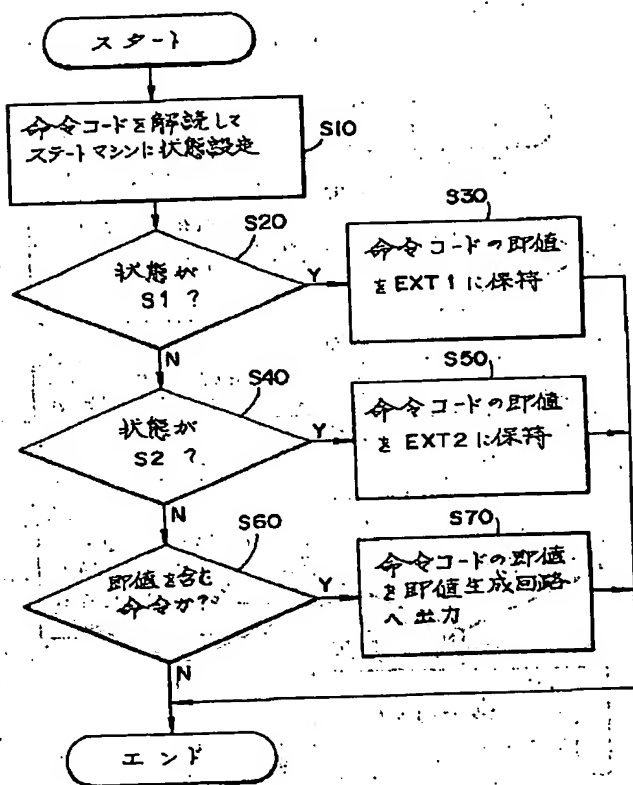
【図1】



【図3】

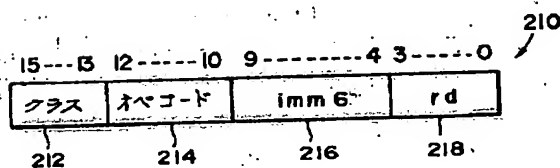


【図4】

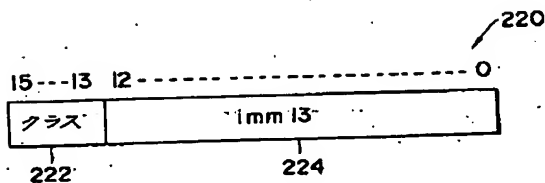


【図7】

(A)

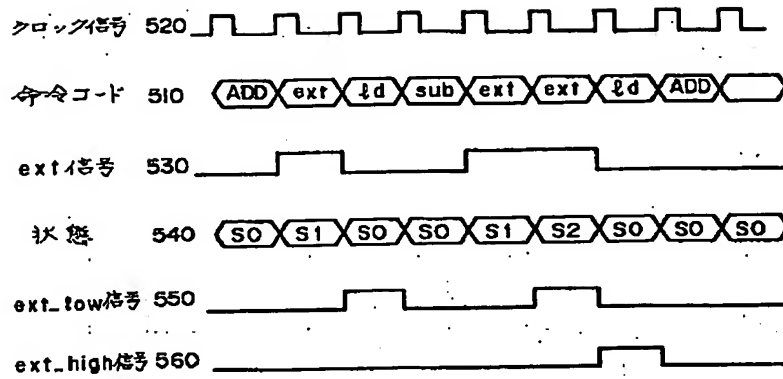


(B)

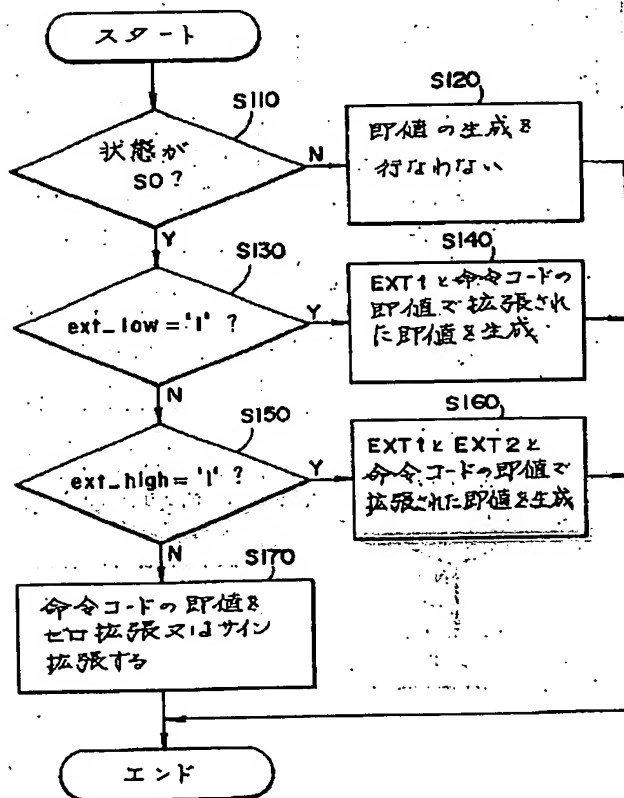


(18)

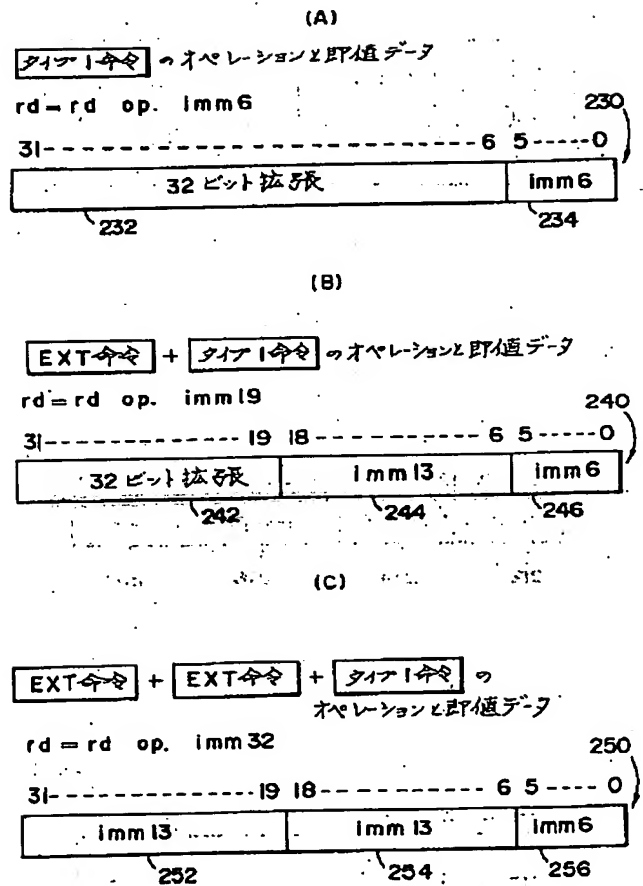
【図5】



【図6】

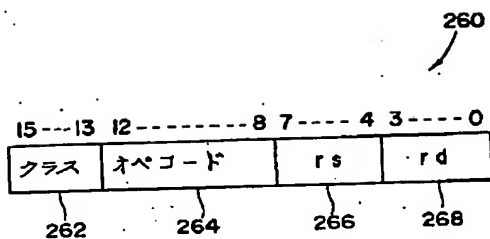


【図8】



(19)

【図9】



【図10】

(A)

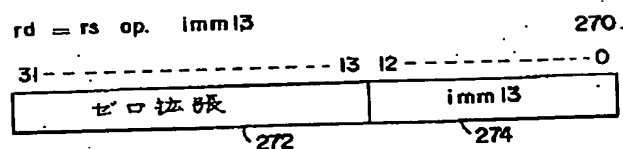
タイプ2命令のオペレーション

$rd = rs \text{ op. } rd$

(B)

EXT命令 + タイプ2命令のオペレーションと即値データ

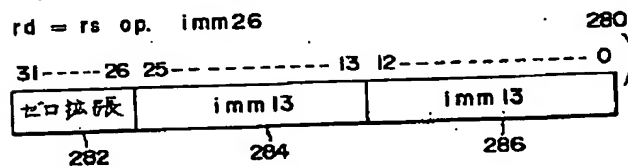
$rd = rs \text{ op. } imm13$



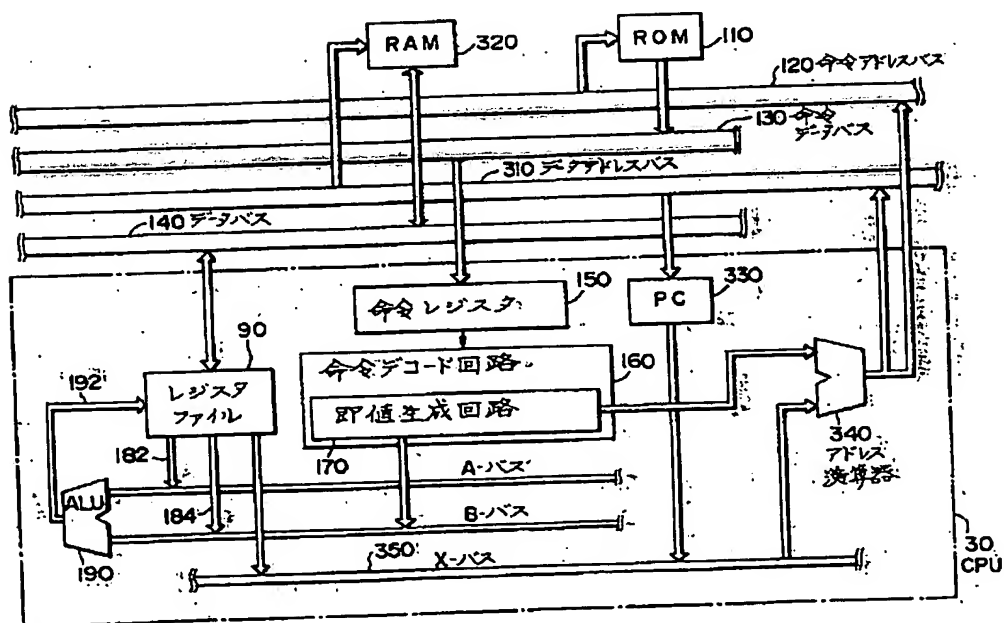
(C)

EXT命令 + EXT命令 + タイプ2命令のオペレーションと即値データ

$rd = rs \text{ op. } imm26$

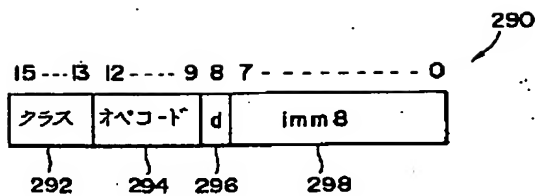


【図11】

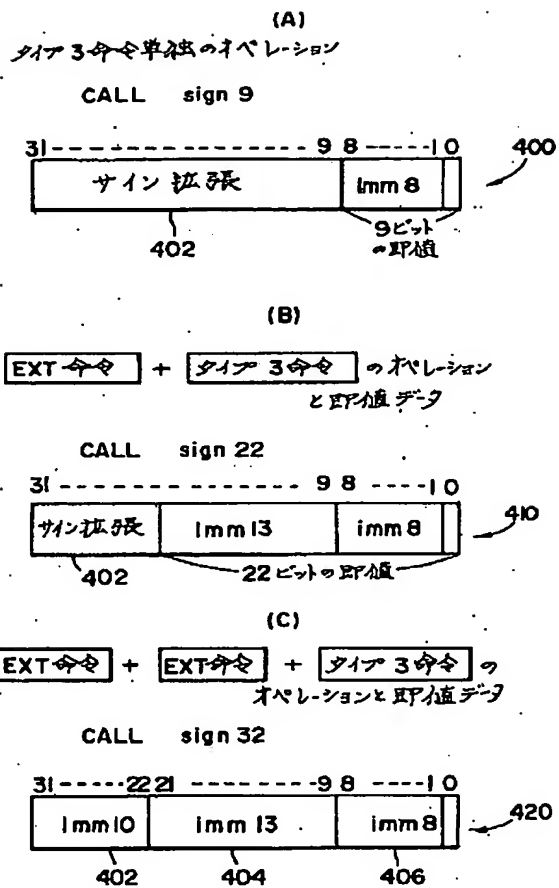


(20)

【図12】

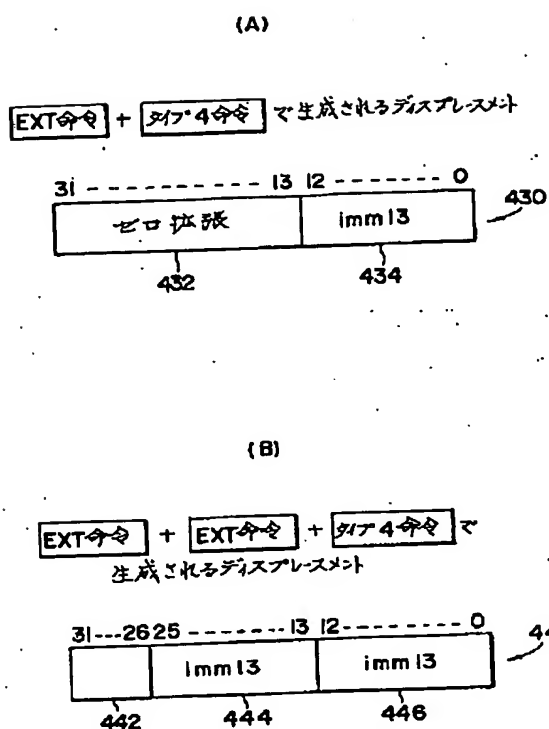


【図13】

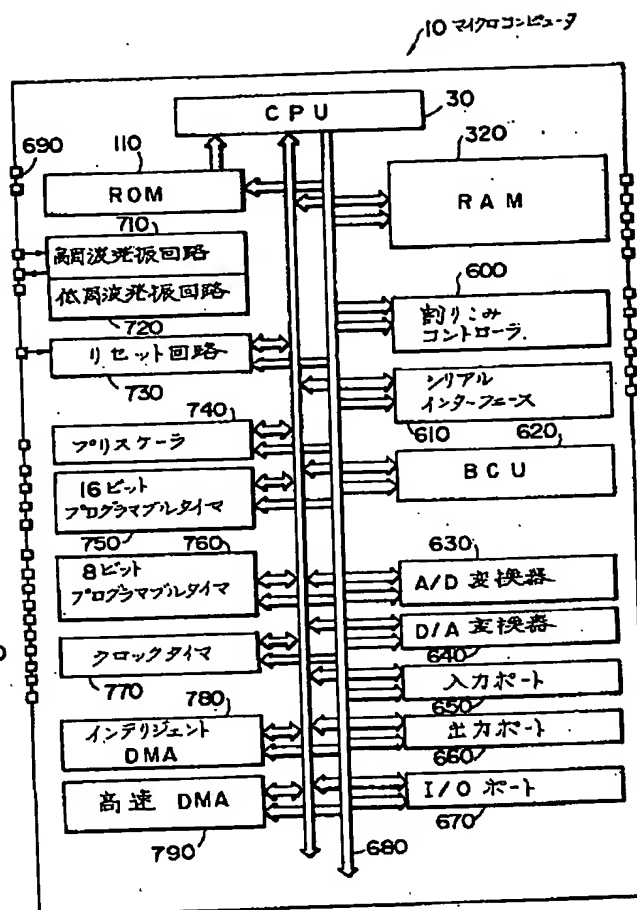


(21)

【図14】



【図15】



【公報種別】特許法第17条の2の規定による補正の掲載
【部門区分】第6部門第3区分
【発行日】平成15年2月14日(2003.2.14)

【公開番号】特開平9-231070
【公開日】平成9年9月5日(1997.9.5)
【年通号数】公開特許公報9-2311
【出願番号】特願平8-353167
【国際特許分類第7版】

G06F 9/30 310
350
9/305

【FI】
G06F 9/30 310 E
350 A
340 F

【手続補正書】

【提出日】平成14年11月14日(2002.11.14)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】発明の名称

【補正方法】変更

【補正内容】

【発明の名称】 情報処理回路

【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

【補正内容】

【特許請求の範囲】

【請求項1】 所与のターゲット命令と、該ターゲット命令の機能を拡張するためのプリフィックス命令が入力される情報処理回路であって、前記ターゲット命令及びプリフィックス命令を入力し、該命令コードのオペレーション内容を解析する命令コード解析手段と、前記命令コード解析手段が解析したオペレーション内容に基づき命令を実行する命令実行手段とを含み、前記命令コード解析手段は、前記プリフィックス命令に基づき、該プリフィックス命令の機能拡張の対象となるターゲット命令の実行に必要な即値を拡張する即値拡張手段を含むよう構成され、前記命令実行手段は、前記即値拡張手段により拡張された即値に基づきターゲット命令を実行することを特徴とする情報処理回路。

【請求項2】 請求項1において、固定長の命令コードを入力して実行するよう構成されていることを特徴とする情報処理回路。

【請求項3】 請求項2において、前記固定長命令コードのビット幅が、前記情報処理回路が処理出来るデータ又はアドレスのビット幅以下であることを特徴とする情報処理回路。

【請求項4】 請求項3において、前記情報処理回路は、16のビット幅の固定長の命令コードを入力するよう構成され、前記即値拡張手段は、前記プリフィックス命令に基づき、前記ターゲット命令の実行に必要な即値を32ビットの幅に拡張し、前記命令実行手段は、前記即値拡張手段で拡張された32ビットの即値を用いてターゲット命令を実行する算術論理演算手段を含むことを特徴とする情報処理回路。

【請求項5】 請求項1～請求項4のいずれかにおいて、

前記即値拡張手段は、プリフィックス命令の命令コードの即値及び前記ターゲット命令の命令コードの即値に基づき、前記ターゲット命令の命令コードに含まれる即値を拡張することを特徴とする情報処理回路。

【請求項6】 請求項1～請求項5のいずれかにおいて、

前記即値拡張手段は、プリフィックス命令の命令コードに含まれた即値を用いてターゲット命令の実行に使用するデータを拡張することを特徴とする情報処理回路。

【請求項7】 請求項1～請求項6のいずれかにおいて、

前記即値拡張手段は、プリフィックス命令の命令コードに含まれた即値を用いてターゲット命令の実行に使用するアドレスを拡張する

(2)

ことを特徴とする情報処理回路。

【請求項8】 請求項1～請求項7のいずれかにおいて、

前記即値拡張手段は、

複数のプリフィックス命令の命令コードに含まれた各即値を用いて、該プリフィックス命令の機能拡張の対象となるターゲット命令の実行に必要な即値を拡張することを特徴とする情報処理回路。

【請求項9】 請求項8において、

前記複数のプリフィックス命令の各命令コードは、プリフィックス命令であることを判別するための共通のコードをそれぞれ含むことを特徴とする情報処理回路。

【請求項10】 請求項8又は請求項9のいずれかにおいて、前記即値拡張手段は、

プリフィックス命令の連続入力回数に基づき決定される処理状態を記憶する処理状態記憶手段と、

前記処理状態記憶手段に記憶された処理状態に基づき、入力されたプリフィックス命令の即値を第1のレジスタ～第mのレジスタ（mは1以上の整数）に保持する前記データ保持手段と、

前記データ保持手段の第1のレジスタ～第mのレジスタ（mは1以上の整数）に保持された即値に基づき、前記ターゲット命令の命令の実行に必要な即値を拡張して生成する手段とを含むことを特徴とする情報処理回路。

【請求項11】 請求項1～請求項10のいずれかにおいて、

前記命令コード解析手段は、

プリフィックス命令入力後にターゲット命令を入力した場合、該ターゲット命令のオペレーション内容を拡張して解釈するターゲット命令機能拡張手段を含み、

前記命令実行手段は、

前記ターゲット命令機能拡張手段で拡張されたオペレーション内容で該ターゲット命令を実行することを特徴とする情報処理回路。

【請求項12】 請求項11において、

前記ターゲット命令機能拡張手段は、

プリフィックス命令入力後に2オペランド命令である所定のターゲット命令を入力した場合、該ターゲット命令の2つのオペランドとプリフィックス命令に含まれた即値を用いて、該ターゲット命令のオペレーション内容を3オペランド命令に拡張して解釈することを特徴とする情報処理回路。

【請求項13】 請求項11において、

前記ターゲット命令機能拡張手段は、

プリフィックス命令入力後にレジスタ指定値を有する所定のターゲット命令を入力した場合、プリフィックス命令の命令コードに含まれた即値に基づき、ターゲット命令の命令コードに指定されたレジスタに格納されたデータ又はアドレスを修正して実行するようにオペレーション内容を拡張することを特徴とする情報処理回路。

2

【請求項14】 請求項13において、

前記ターゲット命令機能拡張手段は、

プリフィックス命令入力後にレジスタ指定値を有する所定のターゲット命令を入力した場合、プリフィックス命令の命令コードに含まれた即値に基づき、ターゲット命令の命令コードに指定されたレジスタに格納されたアドレスのディスプレイースメントを作成し、該ディスプレイースメントを用いて実行するようにオペレーション内容を拡張することを特徴とする情報処理回路。

10 【請求項15】 請求項1～請求項14のいずれかにおいて、

前記命令コード解析手段は、

プリフィックス命令の連続入力回数に基づき、プリフィックス命令のオペレーション内容を拡張するプリフィックス命令機能拡張手段を含むことを特徴とする情報処理回路。

【手続補正3】

【補正対象書類名】明細書

【補正対象項目名】0053

20 【補正方法】変更

【補正内容】

【0053】本発明は、請求項1～請求項15のいずれかにおいて、前記命令コード解析手段に入力する命令コードを記憶する命令コード記憶手段さらに含むことを特徴とする。

【手続補正4】

【補正対象書類名】明細書

【補正対象項目名】0055

【補正方法】変更

30 【補正内容】

【0055】本発明は、上記記載の情報処理回路を含むことを特徴とする半導体集積回路装置である。

【手続補正5】

【補正対象書類名】明細書

【補正対象項目名】0057

【補正方法】変更

【補正内容】

【0057】本発明は、上記記載の情報処理回路を含むことを特徴とするマイクロコンピュータである。

40 【手続補正6】

【補正対象書類名】明細書

【補正対象項目名】0059

【補正方法】変更

【補正内容】

【0059】本発明は、RISC方式であることを特徴とするマイクロコンピュータである。

【手続補正7】

【補正対象書類名】明細書

【補正対象項目名】0061

50 【補正方法】変更

(3)

3

【補正内容】

【0061】本発明は、上記記載のマイクロコンピュー

4

タを用いて制御することを特徴とする電子機器である。